

JANUSZ SOBECKI

Biblioteka Główna i OINT  
Politechniki Wrocławskiej



100100258200

# REKOMENDACJA

## INTERFEJSU UŻYTKOWNIKA W ADAPTACYJNYCH WEBOWYCH SYSTEMACH INFORMACYJNYCH







**Janusz Sobecki**

# **Rekomendacja interfejsu użytkownika w adaptacyjnych webowych systemach informacyjnych**



**Oficyna Wydawnicza Politechniki Wrocławskiej  
Wrocław 2009**

NS 149571

Recenzenci  
Piotr JĘDRZEJOWICZ  
Ngoc Thanh NGUYEN

Opracowanie redakcyjne i korekta  
Alina KACZAK

Projekt okładki  
Dominika OSADCÓW

© Copyright by Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław 2009

Oficyna Wydawnicza Politechniki Wrocławskiej  
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław  
[www.oficyna.pwr.wroc.pl](http://www.oficyna.pwr.wroc.pl)



ISBN 978-83-7493-465-7

340478 L/1

Drukarnia Oficyny Wydawniczej Politechniki Wrocławskiej. Zam. 495/2009

Alc 2009/0439 /K



*Mojej żonie*



## Spis treści

|                                                                                   |    |
|-----------------------------------------------------------------------------------|----|
| 1. Wstęp .....                                                                    | 7  |
| 2. Interakcyjne systemy webowe .....                                              | 11 |
| 2.1. Systemy interakcyjne .....                                                   | 12 |
| 2.1.1. Style interakcji .....                                                     | 13 |
| 2.1.2. Wybór stylu interakcji .....                                               | 18 |
| 2.2. Hipertekst i hipermedia .....                                                | 18 |
| 2.3. Technologie webowe .....                                                     | 21 |
| 2.3.1. Technologie warstwy prezentacji .....                                      | 23 |
| 2.3.2. Technologie warstwy aplikacji .....                                        | 27 |
| 2.3.3. Technologie warstwy bazy danych .....                                      | 30 |
| 2.4. Użyteczność i dostępność systemów interakcyjnych .....                       | 31 |
| 2.4.1. Normy i definicje użyteczności i dostępności .....                         | 31 |
| 2.4.2. Metody zapewnienia użyteczności .....                                      | 34 |
| 2.4.3. Miary użyteczności systemów interakcyjnych .....                           | 44 |
| 2.5. Personalizacja systemów webowych .....                                       | 46 |
| 2.6. Perspektywy rozwoju systemów webowych .....                                  | 49 |
| 2.7. Podsumowanie .....                                                           | 51 |
| 3. Modelowanie użytkownika w webowych systemach rekomendacyjnych .....            | 55 |
| 3.1. Dziedziny zastosowań systemów rekomendacyjnych .....                         | 56 |
| 3.2. Cele i zadania użytkownika w systemie rekomendacyjnym .....                  | 57 |
| 3.3. Zawartość, reprezentacja, tworzenie i aktualizacja profilu użytkownika ..... | 59 |
| 3.3.1. Zawartość profilu .....                                                    | 60 |
| 3.3.2. Reprezentacja profilu użytkownika .....                                    | 63 |
| 3.3.3. Inicjowanie profilu .....                                                  | 67 |
| 3.3.4. Techniki uczenia profilu .....                                             | 69 |
| 3.3.5. Informacja zwrotna od użytkownika .....                                    | 75 |
| 3.3.6. Techniki adaptacji profilu .....                                           | 76 |
| 3.4. Podsumowanie .....                                                           | 77 |
| 4. Wykorzystanie modelu użytkownika w systemach rekomendacyjnych .....            | 81 |
| 4.1. Metoda filtrowania informacji .....                                          | 81 |
| 4.1.1. Filtrowanie demograficzne (DF) .....                                       | 81 |
| 4.1.2. Filtrowanie bazujące na zawartości (CBF) .....                             | 83 |
| 4.1.3. Filtrowanie kolaboratywne (CF) .....                                       | 83 |
| 4.1.4. Inne metody bazujące na filtrowaniu kolaboratywnym .....                   | 85 |
| 4.1.5. Wnioskowanie oparte na przypadkach użycia (CBR) .....                      | 86 |
| 4.1.6. Podejście hybrydowe (HA) .....                                             | 87 |
| 4.2. Techniki dopasowania rekomendowanego elementu do profilu użytkownika .....   | 88 |
| 4.2.1. Dokładne dopasowanie na podstawie słów kluczowych .....                    | 88 |

|                                                                                                                      |     |
|----------------------------------------------------------------------------------------------------------------------|-----|
| 4.2.2. Funkcje odległości i podobieństwa . . . . .                                                                   | 89  |
| 4.2.3. Klasyfikacja . . . . .                                                                                        | 97  |
| 4.3. Dopasowanie pomiędzy profilami . . . . .                                                                        | 98  |
| 4.3.1. Znajdowanie podobnych użytkowników . . . . .                                                                  | 98  |
| 4.3.2. Utworzenie sąsiedztwa . . . . .                                                                               | 100 |
| 4.3.3. Wyznaczenie predykcji na podstawie wybranego sąsiedztwa . . . . .                                             | 100 |
| 4.4. Metody oceny efektywności systemów rekomendacyjnych . . . . .                                                   | 102 |
| 4.4.1. Miary efektywności wywodzące się z IR . . . . .                                                               | 104 |
| 4.4.2. Pozostałe miary badania efektywności systemów rekomendacyjnych . . . . .                                      | 110 |
| 4.4.3. Wykorzystanie miar efektywności do badania rekomendacji w zbiorze MovieLens . . . . .                         | 114 |
| 4.5. Podsumowanie . . . . .                                                                                          | 116 |
| 5. Adaptacja interfejsów użytkownika metodą consensusu . . . . .                                                     | 119 |
| 5.1. Architektura adaptacyjnego interfejsu użytkownika webowego systemu informacyjnego . . . . .                     | 120 |
| 5.2. Teoria consensusu . . . . .                                                                                     | 123 |
| 5.2.1. Pojęcia podstawowe . . . . .                                                                                  | 124 |
| 5.3. Definicja systemu konfliktu . . . . .                                                                           | 125 |
| 5.3.1. Profile konfliktu . . . . .                                                                                   | 126 |
| 5.3.2. Wyznaczenie consensusu dla profili użytkowników . . . . .                                                     | 127 |
| 5.4. Profil użytkownika . . . . .                                                                                    | 129 |
| 5.5. Wykorzystanie profilu użytkownika . . . . .                                                                     | 133 |
| 5.6. Badania eksperymentalne systemu <i>Megane</i> . . . . .                                                         | 135 |
| 5.6.1. Przebieg eksperymentu . . . . .                                                                               | 136 |
| 5.6.2. Wyznaczenie efektywności rekomendacji interfejsu użytkownika . . . . .                                        | 138 |
| 5.7. Podsumowanie . . . . .                                                                                          | 143 |
| 6. Adaptacja interfejsu użytkownika z wykorzystaniem metod inspirowanych naturą . . . . .                            | 145 |
| 6.1. Wykorzystanie AIS w adaptacji interfejsów użytkownika . . . . .                                                 | 146 |
| 6.1.1. Model AIS zastosowany w RS . . . . .                                                                          | 146 |
| 6.1.2. Architektura systemu <i>Reporter</i> . . . . .                                                                | 148 |
| 6.1.3. Implementacja i weryfikacja . . . . .                                                                         | 149 |
| 6.2. Wykorzystanie metafory kolonii mrówek w adaptacji interfejsów użytkownika . . . . .                             | 151 |
| 6.2.1. Architektura systemu rekomendacyjnego wykorzystująca ontologie . . . . .                                      | 152 |
| 6.2.2. Metafora kolonii mrówek . . . . .                                                                             | 155 |
| 6.2.3. Badania efektywności i własności wykorzystania kolonii mrówek w rekomendacji interfejsów użytkownika. . . . . | 158 |
| 6.3. Podsumowanie . . . . .                                                                                          | 161 |
| 7. Wykorzystanie klasyfikacji przybliżonej do wyznaczenia optymalnego profilu użytkownika . . . . .                  | 163 |
| 7.1. Koncepcja i podstawowe pojęcia klasyfikacji przybliżonej . . . . .                                              | 163 |
| 7.2. Klasyfikacja przybliżona profili użytkowników . . . . .                                                         | 166 |
| 7.3. Rozwiązania problemów klasyfikacji przybliżonej . . . . .                                                       | 168 |
| 7.3.1. Rozwiązanie problemu RC-1 . . . . .                                                                           | 169 |
| 7.3.2. Rozwiązanie Problemu RC-2 . . . . .                                                                           | 170 |
| 7.3.3. Rozwiązanie Problemu RC-3 . . . . .                                                                           | 171 |
| 7.4. Przykład rozwiązania problemów RC-1, RC-2 i RC-3 . . . . .                                                      | 172 |
| 7.5. Badania eksperymentalne klasyfikacji przybliżonej . . . . .                                                     | 174 |
| 7.6. Podsumowanie . . . . .                                                                                          | 177 |
| 8. Uwagi końcowe . . . . .                                                                                           | 179 |
| 9. Literatura . . . . .                                                                                              | 181 |

# 1. Wstęp

System World Wide Web (WWW), zwany powszechnie webem, jest obecnie najpopularniejszym uzupełnieniem całego „repertuaru” współczesnych mediów, zwłaszcza ze względu na dużą efektywność i szeroki zakres zastosowania w tak różnorodnych branżach, jak biznes, edukacja, kultura i nauka [Wu 2005].

Liczba użytkowników systemów webowych zarówno w przeszłości [Taylor 2002], jak i obecnie wzrasta [Miniwatts 2008]. Już wielokrotnie z roku na rok ich liczba podwajała się, a nawet potrajała. Od kilku lat w wielu wysoko rozwiniętych krajach, m.in. w Islandii, Holandii, Dani, USA i Korei ich liczba osiągnęła praktycznie maksymalny pułap [Miniwatts 2008]. Jednak w zdecydowanej większości nieco mniej technologicznie rozwiniętych państw, w tym i w Polsce, ich liczba będzie zwiększała się jeszcze przez wiele lat, niezależnie od tego, iż wzrost w ostatnich ośmiu latach był już i tak bardzo imponujący, gdyż sięgał nawet powyżej 400% rocznie [Miniwatts 2008].

Wraz ze wzrostem liczby użytkowników systemów webowych zwiększa się ich zróżnicowanie, wynikające z odmiennej płci, wieku, wykształcenia, zainteresowań, ale także wynikające z różnic kulturowych, społecznych, narodowych i antropologicznych. Różnice te oznaczają także odmienne potrzeby i oczekiwania użytkowników w odniesieniu do systemów webowych, a zwłaszcza do sposobu komunikowania się z nim, czyli konfiguracji interfejsu [Stary 2001].

Systemy webowe ze względu na swoją wszechstronność są obecnie wykorzystywane w rozlicznych dziedzinach życia społecznego i gospodarczego, dlatego systemy te znacznie różnią się między sobą i to nie tylko ze względu na różnorodne dziedziny zastosowania, ale także ze względu na wielorakość realizowanych przez nie celów, wśród których można wymienić informowanie, promowanie (reklama) lub edukowanie.

Systemy webowe różnią się także ze względu na niejednorodne platformy uruchomieniowe charakteryzujące się zróżnicowanymi parametrami przetwarzania, prezentacji i wprowadzania danych. Systemy webowe są wykorzystywane na komputerach klasy PC (desktop i notebook), Mac (desktop i notebook) i stacjach roboczych pracujących w systemie Unix, przez kioski informacyjne, web-TV, przeglądarki zainstalowane w samochodach i samolotach, a także w urządzeniach typu PDA oraz w telefonach komórkowych.

Wielostronne zróżnicowanie systemów webowych wynikające z ich różnorodnego przeznaczenia, zastosowanych platform oraz zróżnicowanych użytkowników powoduje, że zaprojektowanie takiego interfejsu użytkownika systemu webowego, który byłby



wystarczająco użyteczny [ISO 1997] we wszystkich przypadkach, jest praktycznie niemożliwe [Sobecki 2001].

Tradycyjnie problemami interfejsów użytkownika w systemach komputerowych, w tym także interakcją w systemach webowych zajmuje się nauka interakcji człowiek–komputer (ang. *Human–Computer Interaction* – HCI). Celem podstawowym HCI jest tworzenie takich systemów interakcyjnych, które można, na podstawie odpowiednich norm, np. ISO, uznać za użyteczne [ISO 1997]. Metodologia projektowania systemów interakcyjnych wypracowana w ramach HCI [Newman 1996], nakazuje projektowanie interfejsu dostosowanego do: celu aplikacji, docelowej grupy użytkowników oraz platformy systemowo-sprzętowej [Sobecki 2000a]. Dlatego też obecne systemy webowe coraz częściej wyposażone są w mechanizmy umożliwiające adaptację interfejsu do wszystkich lub choćby do części z wymienionych czynników. Systemy tego typu nazywa się spersonalizowanymi [Kobsa 2001] lub adaptacyjnymi [Stephanidis 1998]. Bardzo często w tego typu systemach wykorzystuje się technologie opracowane w ramach dziedziny naukowej systemów rekomendacyjnych (ang. *Recommendation Systems* – RS) [Montaner 2003]. Efektywność tych technologii jest badana z użyciem specyficznej techniki [Herlocker 2004], której część wywodzi się z techniki opracowanej wiele lat temu na gruncie wyszukiwania informacji (ang. *Information Retrieval* – IR) [Salton 1975] i [Rijsbergen 1979].

Celem niniejszej monografii jest zaprezentowanie wybranych wyników opracowanych przez Autora w zakresie rekomendacji interfejsu użytkownika w adaptacyjnych systemach webowych. Zastosowanie tych metod umożliwia uzyskanie wzrostu użyteczności w systemów webowych w porównaniu z systemami, które nie wykorzystują rekomendacji interfejsu użytkownika. Rozwiązywanie problemów w niniejszym zakresie wymaga zastosowania metod opracowanych przez przynajmniej kilka różnych dziedzin naukowych, takich jak interakcja człowiek–komputer, systemy rekomendacyjne, systemy webowe, użyteczność oraz wykorzystuje liczne metody opracowane w ramach wyszukiwania informacji, sztucznej inteligencji i teorii wyboru. Konieczne jest przedstawienie podstawowych pojęć oraz metod rozwiązania wybranych problemów wywodzących się z tych dziedzin. W ramach pracy przedstawiono zarówno motywację do zajęcia się tą problematyką, jak i trzy metody rekomendacji interfejsu użytkownika oraz metodę umożliwiającą poprawę jakości modelowania użytkownika w systemie z rekomendacją interfejsu.

Monografia obejmuje oprócz wstępu siedem rozdziałów. W rozdziale drugim przedstawiono problematykę interakcyjnych systemów webowych. W ramach tego rozdziału przedstawiono definicję interakcji i systemów interakcyjnych, podstawowe style interakcji oraz zasady ich wyboru. Zaprezentowano historyczne źródła systemów webowych, a mianowicie hipertekst i hipermedia. Kolejno przedstawiono technologie webowe, ze szczególnym uwzględnieniem technologii warstwy prezentacji, które umożliwiają tworzenie systemów spersonalizowanych. W dalszej kolejności przedstawiono definicje, normy i metody zapewnienia użyteczności systemów interakcyjnych oraz stosowane miary użyteczności służące do wyznaczenia jakości lub porównania różnych systemów interakcyjnych, ze szczególnym uwzględnieniem webowych systemów interakcyj-

nych. Przedstawiono również rodzaje i zastosowania personalizacji systemów webowych oraz perspektywy rozwoju systemów webowych.

Problemy modelowania użytkowników w rekomendacyjnych systemach webowych zawiera rozdział 3. Podano obszary zastosowania systemów rekomendacyjnych oraz cele i zadania użytkowników systemów rekomendacyjnych. Przedstawiono metody reprezentacji profilu użytkownika, jego tworzenia i utrzymania w systemie rekomendacyjnym. Oprócz metod znanych z literatury przedstawiono wyniki prac Autora w tym zakresie.

Metody wykorzystania profilu użytkownika w systemie rekomendacyjnym przedstawiono w rozdziale czwartym. W ramach rozdziału przedstawiono metody filtrowania informacji, techniki dopasowania rekomendowanego elementu do profilu użytkownika, techniki dopasowania pomiędzy profilami oraz metody oceny efektywności systemów rekomendacyjnych. Zaprezentowano również wyniki przeprowadzonych przez Autora badań efektywności metod wyznaczania dopasowania pomiędzy profilami użytkowników oraz wybranych metod filtrowania informacji.

Metodę adaptacji interfejsu użytkownika w systemie webowym z wykorzystaniem przystosowanej do tego celu przez Autora metody wyznaczania consensusu przedstawiono w rozdziale piątym. Rozdział zawiera opis opracowanej przez Autora architektury adaptacji interfejsu użytkownika webowego systemu informacyjnego, teorii consensusu wraz z wprowadzonymi przez autora modyfikacjami, opis profilu użytkownika i jego wykorzystania w systemie z adaptacją interfejsu, oraz krótki opis implementacji systemu i wyniki badań efektywności systemu *Megane*.

W rozdziale szóstym przedstawiono dwie metody bazujące na naturze, które przystosowano przez Autora do tworzenia adaptacyjnych interfejsów użytkownika. Pierwsza z metod rekomendacji interfejsu użytkownika bazuje na sztucznym systemie immunologicznym (ang. *Artificial Immune System* – AIS). Zaprezentowano metodę AIS, jej implementację do rekomendacji artykułów z wiadomościami i rekomendacji interfejsu w systemie *Reporter*, oraz wyniki badań nad efektywnością zastosowania tej metody do rekomendacji artykułów i interfejsu użytkownika. Druga część rozdziału zawiera opis wykorzystania metafory kolonii mrówek (ang. *Ant Colony Metaphor* – ACM). W ramach tego opisu przedstawiono własną architekturę systemu rekomendacyjnego wykorzystującą ontologię, zaprezentowano podstawy ACM oraz jej zastosowanie do rekomendacji interfejsu wraz z analizą zastosowania tej metody z wykorzystaniem danych testowych ze zbioru *MovieLens* [Grouplens 2008].

Zastosowanie klasyfikacji przybliżonej do wyznaczania optymalnej zawartości profilu użytkownika przedstawiono w rozdziale siódmym. Zaprezentowano podstawowe pojęcia klasyfikacji przybliżonej, następnie klasyfikację przybliżoną profili użytkownika, przedstawienie problemów klasyfikacji przybliżonej profili ich różnych rozwiązań, w tym także z wykorzystaniem algorytmów genetycznych, a także właściwości klasyfikacji i ich dowody. Podano wyniki przeprowadzonych przez Autora badań klasyfikacji przybliżonej profili użytkowników systemu *Megane* oraz danych ze zbioru *Adult* [Adult 1998].

Rozdział ósmy zawiera podsumowanie zaprezentowanych w monografii tez oraz perspektywy rozwoju rekomendacji interfejsów użytkownika w webowych systemach adaptacyjnych.



## 2. Interakcyjne systemy webowe

System WWW jest obecnie najważniejszą usługą informacyjną sieci Internet. Sieć Internet powstała w 1969 roku, jako rezultat 10-letnich badań agencji ARPA (ang. *Advanced Research Project Agency* – przemianowanej w 1972 roku na DARPA, ang. *Defense Advanced Research Project Agency*) Departamentu Obrony Rządu USA [Knuckles 2005]. Przedsięwzięcie, mające w zamyśle cele obronne, zyskało z czasem zainteresowanie środowisk naukowych [Sobecki 1999b], a następnie biznesowych. Pierwszą i nadal powszechnie używaną usługą sieci Internet była poczta elektroniczna (email), następnie zdalny terminal (telnet), który jest znany wyłącznie w kręgach informatyków i w końcu ftp (ang. *file transfer protocol*) – usługa narzędziowa przeznaczona do transmisji plików. W latach 80. ubiegłego stulecia powstało bardzo wiele różnorodnych usług, obecnie już całkiem zapomnianych i to pomimo ich dużej popularności w określonym czasie, np. usługa informacyjna Gopher, usługa indeksująca Archie czy też katalogowa X.500.

Koncepcja WWW została opracowana w 1989 roku w międzynarodowym instytucie badań jądrowych CERN w Genewie przez Tima Bernersa-Lee (obecnie prezesa W3 Consortium). Początkowo, po jej przedstawieniu szefom CERN-u oraz pewnym szerszym kręgom, nie spotkała się z większym zainteresowaniem. W październiku 1990 przy pomocy Roberta Cailliau wstępna koncepcja została przeformułowana oraz wybrano dla niej nazwę: World Wide Web [Feizabadi 1998]. System WWW został oparty na idei wykorzystania hipertekstu, a właściwie hipermediów.

Idea globalnego systemu informacyjnego, gdzie stosuje się hipertekst, powstała przede wszystkim na podstawie doświadczeń z wykorzystania hipertekstu i hipermediów w latach 80. XX wieku do celów wspomagania nauczania. Z czasem wykorzystywano jego możliwości do prezentowania wszelkich niekomercyjnych informacji z dziedziny edukacji, nauki, kultury i turystyki [Koćcio 1995]. Mniej więcej w połowie lat 90. ubiegłego wieku zaakceptowano wykorzystanie Internetu w biznesie, co zaowocowało powstaniem wielu serwisów komercyjnych, takich jak portale, zarabiające głównie na reklamach lub płatne specjalistyczne serwisy informacje, np. giełdowe.

Systemy webowe są podobnie jak hipertekst czy hipermedia systemami interakcyjnymi, dlatego w pracy podano definicję interakcji i systemów interakcyjnych, podstawowe style interakcji oraz zasady ich wyboru. Przedstawiono technologie webowe, począwszy od podstawowych HTML i HTTP, poprzez technologie trójwarstwowe, po współczesne aplikacje RIA [Allaire 2002]. Odpowiednią efektywność systemów webo-

wych realizuje się przede wszystkim przez zastosowanie metod zapewnienia użyteczności, dlatego przedstawiono problemy użyteczności i dostępności systemów webowych, gdzie przytoczono ich definicje oraz regulujące je normy oraz wybrane metody, które umożliwiają ocenę użyteczności, a zwłaszcza adaptacyjnych interfejsów systemów webowych. W przedostatnim punkcie rozdziału przedstawiono zalety oraz możliwości personalizacji systemów webowych oraz perspektywę rozwoju systemów webowych.

## 2.1. Systemy interakcyjne

Podstawową cechą systemów interakcyjnych jest wzajemna wymiana informacji pomiędzy użytkownikiem a systemem komputerowym. Tradycyjnie użytkownikiem tych systemów jest człowiek, i tylko takie systemy są przedmiotem rozważań w niniejszej pracy, jednak w literaturze można znaleźć również przykłady systemów, których użytkownikami są zwierzęta domowe, takie jak koty [Iyer 2005] lub małpy człekokształtne [Age Company 2007].

W *Słowniku Encyklopedycznym – Informatyka* przedstawiono definicję *interakcja człowiek – komputer*, jako „tryb pracy polegający na naprzemiennej wymianie informacji między użytkownikiem a komputerem”, w którym „użytkownik wydaje komputerowi polecenia, których skutki może na bieżąco obserwować i podejmować uzależnione od nich decyzje” [Płoski 2001]. W innej bardzo popularnej definicji Sheizafa Rafaeliego interakcyjność wyraża poziom wzajemnej zależności pomiędzy komunikatami w pewnej serii komunikatów [Rafaeli 1988]. Od tego procesu można również wywieść cechę interakcyjności, która może przyjmować następujące trzy wartości:

- zupełnie nieinterakcyjna, gdy wiadomość nie zależy od poprzednich,
- reakcyjna, gdy wiadomość zależy od bezpośrednio poprzedzającej wiadomości,
- interakcyjna, gdy wiadomość zależy od pewnej grupy poprzednich wiadomości oraz związków zachodzących pomiędzy nimi.

Na podstawie pojęć interakcji można zdefiniować centralne pojęcia do niniejszej pracy, a mianowicie system interakcyjny oraz interakcja człowiek–komputer. Systemy interakcyjne można zdefiniować jako klasę systemów, których działanie wymaga znacznego stopnia interakcji z użytkownikiem [Kotonya 1998]. Interakcja ta może być realizowana na wiele różnorodnych sposobów, które z kolei wymagają wykorzystania różnorodnych urządzeń wejścia–wyjścia. Metody interakcji omówiono w podpunkcie dotyczącym stylów interakcji.

Interakcja człowiek–komputer (HCI) jest nazwą nauki, która zajmuje się metodami projektowania, tworzenia i badania systemów interakcyjnych ze szczególnym uwzględnieniem czynnika ludzkiego, tj. aspektów psychologicznych i socjologicznych odnoszących się do użytkowników systemów komputerowych. Termin HCI został zaadaptowany w połowie lat 80. ubiegłego wieku [Preece 1994], w czasie gdy systemy komputerowe zyskały nowe rzesze użytkowników oraz zostały zastosowane w wielu dziedzinach gospodarki i edukacji. W ówczesnych czasach pojęcie HCI rozumiano za Beckerem

i Buxtonem jako pewien zbiór procesów, dialogów i akcji wykorzystywanych przez użytkownika do interakcji z komputerem. W latach dziewięćdziesiątych XX w. zaadaptowano szerszą definicję, prezentowaną również w pracy [Hewett 1992], określającą HCI jako dyscyplinę zajmującą się projektowaniem, ewaluacją i implementacją interakcyjnych systemów komputerowych przeznaczonych dla ludzi oraz głównymi zjawiskami z ich otoczenia [Preece 1994].

### 2.1.1. Style interakcji

Style interakcji reprezentują alternatywne strategie projektowania i realizacji interfejsu użytkownika. W pracy [Neman 1996] wyróżniono następujące trzy typy stylów interakcji: klawiszowo-modalny, bezpośredniej manipulacji i lingwistyczny. Wraz z postępem technologicznym wykształciły się nowoczesne metody interakcji, takie jak multimodalne, niebazujące na komendach style interakcji [Nystrand 2001] oraz wirtualna rzeczywistość.

#### 2.1.1.1. Styl klawiszowo-modalny

W typie klawiszowo-modalnym interfejs obsługiwany jest przez pewną liczbę klawiszy (przycisków), które w zależności od kontekstu mogą pełnić różną funkcję, stąd określenie modalność [Raskin 2000]. Do tego typu można zaliczyć następujące podtypy:

- interakcja oparta na menu (system generuje zestaw opcji wywołujących jakieś akcje lub generujących nowe opcje),
- pytania i odpowiedzi (system prezentuje serie pytań w formie tekstowej i oczekuje odpowiedzi wprowadzanej na klawiaturze),
- klawisze funkcyjne (system wykorzystuje pewien zestaw specjalnych klawiszy, a na osobnym urządzeniu wyświetla informację zwrotną),
- komendy prezentowane głosowo (np. w systemach telefonicznych – typu audiotele lub automatyczna sekretarka GSM itp.).

**Menu**, pomimo iż jest już stosowane w systemach interakcyjnych od ponad trzydziestu lat, ciągle należy do najbardziej popularnych stylów interakcji. Wyróżnia się następujące typy menu:

- *pop-up*, które pojawia się wtedy, gdy kliknie się na określony obszar i pozostaje do czasu, gdy użytkownik o tym zadecyduje,
- *pull-down*, które ‘zjeżdża na dół jak roleta’ z górnej belki, a po wyborze opcji automatycznie wraca na swoje miejsce – typ ten jest specjalnym przykładem menu *pop-up*,
- *pie-menu* jest to menu w kształcie ‘placka’, jego zaletą jest równa odległość umieszczenia opcji wyboru od środka ekranu, co powoduje wyrównanie szybkości wyboru każdej opcji oraz zmniejszenie prawdopodobieństwa popełnienia błędu wyboru opcji,
- *hipertekst* jest to menu zawarte wewnątrz dokumentu hipertekstowego.

Menu jest tak popularnym stylem interakcji, że doczekało się licznych opracowań w postaci przewodników stylów opisujących sposoby ich stosowania przez wielu świa-

towych liderów technologicznych, takich jak Apple, Macromedia (obecnie przejęte przez Adobe) czy Microsoft, np. [Neman 1996]:

- stosuj semantykę zadania do tworzenia menu (pojedyncza, sekwencja liniowa, struktura drzewa, cykliczne i acykliczne sieci),
- pokazuj pozycję stosując grafikę, numery czy tytuły,
- stosuj sensowne grupowanie i porządkowanie,
- stosuj krótkie nazwy zaczynające się od słowa kluczowego,
- używaj spójnej gramatyki, wyglądu i terminologii,
- stosuj wyróżnienia i skróty,
- dopuść skoki do poprzedniego i głównego menu,
- rozważ pomoc on-line, mechanizmy selekcji dla początkujących, czas odpowiedzi i wyświetlania, rozmiar ekranu.

Styl **pytań i odpowiedzi** spotyka się w zastosowaniach niskobudżetowych, w których interakcja z użytkownikiem jest niezbyt rozbudowana i dopuszczany jest udział niedoświadczonych użytkowników [Newman 1996]. Styl pytań i odpowiedzi charakteryzuje się następującymi cechami: pytania zadawane są pojedynczo oraz kolejne pytanie może zależeć od odpowiedzi na poprzednie. Ten typ komunikacji sprawdza się w sytuacjach, gdy spodziewane wejście jest ograniczone co do typu (np. kwota pieniędzy, numer części itp.), dlatego najczęściej jest stosowany w bankomatach, a także w katalogach bibliotecznych.

Styl **klawiszowy funkcyjnych** wykorzystuje się w systemach powszechnego użytku (automaty biletowe, automaty sprzedażowe – np. napojów lub batonów, bankomaty, systemy zabezpieczeń, ale także systemy używane w gospodarstwach domowych – pralki, systemy ogrzewania, piloty odbiorników TV). Styl ten wykorzystuje technologię wprowadzania danych opartą na: przyciskach, klawiszach funkcyjnych i klawiaturze numerycznej, ale także stosuje czytniki: kodów kreskowych, kart magnetycznych i chipowych. Projektując interfejs wykorzystujący klawisze funkcyjne należy zadbać o zarządzanie urządzeniami aktywującymi (np. poprzez określenie stanów przejścia) oraz upewnić się, że użytkownik jest świadomy aktualnego stanu systemu przez prezentację odpowiednich zapytań i komunikatów. Brak komunikatów lub ich niejednoznaczność może rodzić wiele problemów użyteczności, a w konsekwencji może zniechęcać użytkowników do korzystania z takich urządzeń.

**Interakcja głosowa** wykorzystywana jest przede wszystkim w systemach, których kanałem komunikacyjnym pomiędzy systemem a użytkownikiem jest głosowa rozmowa telefoniczna. Komunikaty systemu podawane są w formie nagranych uprzednio głosu, złożenia nagranych słów lub, obecnie coraz częściej, w formie syntezy głosu. Główną zaletą stosowania interakcji głosowej w komunikacji z systemem jest możliwość wykorzystania istniejącej sieci telefonicznej, co w czasie przed upowszechnieniem sieci Internet i aplikacji webowych, było bardzo dużą zaletą. Główną wadą tego mechanizmu jest przeładowanie pamięci użytkownika, który zostaje zmuszony do wysłuchania co najmniej kilku komunikatów zanim będzie mógł podjąć ostateczną decyzję o wyborze pożądanej opcji.



### 2.1.1.2. Styl bezpośredniej manipulacji

W stylu bezpośredniej manipulacji interfejs może wyświetlić jednocześnie wiele obiektów naraz, a z pomocą odpowiedniego narzędzia użytkownik może bezpośrednio na nich wykonywać odpowiednie akcje. W tym stylu wyróżnia się następujące podtypy:

- bezpośrednia manipulacja graficzna, realizowana przez tzw. graficzny interfejs użytkownika (ang. *Graphical User Interface* – GUI),
- wypełnianie formularzy, w którym system wyświetla zestaw pól tekstowych, następnie użytkownik wprowadza za pomocą klawiatury odpowiednie wartości).

Styl **bezpośredniej manipulacji graficznej** rozpropagowany w latach 80. ubiegłego stulecia przez firmę Apple, a następnie z jeszcze większym sukcesem komercyjnym wyznaczonym liczbą wdrożeń zastosowany przez firmę Microsoft w dość licznych wersjach systemu Windows, stał się obecnie standardem interakcji człowiek–komputer. Ze względu na postęp technologiczny ten typ komunikacji jest wykorzystywany nie tylko w systemach desktopowych i notebookach, ale także, dzięki zastosowaniu niewielkich ekranów dotykowych, w różnego typu systemach mobilnych, takich jak PDA i tzw. smartfony. Bezpośrednia manipulacja graficzna stwarza użytkownikowi możliwość manipulowania pojedynczymi obiektami, mającymi indywidualną reprezentację graficzną, poprzez zastosowanie odpowiednich akcji (komend) na tych, uprzednio wyselekcjonowanych, obiektach. Obecnie coraz częściej, szczególnie w przypadku gier komputerowych, prezentowane obiekty wyświetlane są w postaci grafiki trójwymiarowej (3D). Manipulacja tak prezentowanymi obiektami może być bardziej efektywna niż manipulacja obiektami prezentowanymi w postaci dwuwymiarowej (2D) [Christmann 2006]. Bezpośrednia manipulacja graficzna wykorzystująca prezentację obiektów w postaci 3D jest elementem koniecznym w prezentowanym dalej stylu wirtualnej rzeczywistości (ang. *Virtual Reality* – VR).

Styl **formularzy** jest ciągle dominującym stylem interakcji w systemach informacyjnych z bazą danych, w których formularze wykorzystywane są przede wszystkim do wprowadzania i wyszukiwania danych. Formularze umożliwiają efektywną realizację złożonych (w porównaniu z interakcją kluczowo-modalną) zadań, takich jak wspomniane już wprowadzanie i wyszukiwanie danych. Ich podstawową wadą jest konieczność przygotowania osobnego formularza dla każdej interakcji.

### 2.1.1.3. Styl lingwistyczny

W stylu lingwistycznym wprowadzanie danych odbywa się za pomocą klawiatury [Newman 1996] lub np. za pomocą czytnika pisma lub mikrofonu, jednak wtedy konieczne jest zastosowanie dodatkowego specjalistycznego oprogramowania do rozpoznawania pisma lub mowy. W stylu tym wyróżnia się następujące podtypy:

- linia komend (użytkownik wprowadza komendę za pomocą klawiatury lub innego urządzenia wejścia, rezultat prezentowany jest użytkownikowi zwykle w postaci tekstowej),
- tekst w języku naturalnym (użytkownik wprowadza tekst w języku naturalnym, który poddawany jest analizie syntaktycznej i semantycznej, która z kolei prowadzi do

odpowiedniej interpretacji oraz odpowiedzi systemu często również w języku naturalnym).

**Linia komend**, oprócz klawiszy funkcyjnych jest jednym ze starszych stylów interakcji. Był on wykorzystywany zwłaszcza w systemach operacyjnych, takich jak UNIX, DOS czy Linux oraz w starszych wersjach systemów bankowych czy rezerwacji lotów. Styl ten ma wiele, ciągle docenianych, zalet, takich jak: prosta technologia ułatwiająca implementację, szeroka gama możliwości (wprowadzanie, wyszukiwanie itp.) oraz szybki sposób wprowadzania i wyszukiwania danych. Styl ten ma jednak jedną istotną wadę, która powoduje, że jest on coraz rzadziej stosowany, a mianowicie wymaga długiego czasu uczenia się konkretnej jej implementacji, co bardzo ogranicza krąg potencjalnych użytkowników.

**Język naturalny** jest dla ludzi najbardziej naturalną formą komunikacji. Obecnie dzięki postępowi w dziedzinach przetwarzaniu języka naturalnego (ang. *Natural Language Processing*) i sztucznej inteligencji (ang. *Artificial Intelligence*) system komputerowy jest w stanie „zrozumieć” nie tylko proste zdania w wielu językach, ale coraz bardziej złożone teksty. Pomimo że interakcja w języku naturalnym wydaje się „panaceum” na wszelkie problemy interakcji, ma ona swoje wady, wśród których należy wymienić: konieczność nauczania użytkownika podzbioru języka rozumianego przez komputer (często łatwiej coś wybrać z zestawu niż słownie opisać pożądany wybór), fakt, iż użytkownik nie do końca jest świadomy możliwości systemu (co do komunikacji i jego funkcji) oraz konieczność wprowadzania w miarę długiego tekstu na klawiaturze. Wydaje się więc, że wykorzystanie języka naturalnego stanie się szersze, gdy oprócz technologii przetwarzania języka naturalnego zastosuje się technologie rozpoznawania głosu, aby umożliwić komunikację za pomocą mowy.

Obecnie w systemach webowych wykorzystuje się różnego typu boty wykorzystujące język naturalny do komunikacji z użytkownikiem. Mają one zastosowanie w systemach pomocy, help-deskach oraz systemach tropiących nielegalne zachowania w sieci Internet.

#### 2.1.1.4. Styl multimodalny

Multimodalne, niebazujące na komendach style interakcji [Nystrand 2001] charakteryzują się ciągłym monitorowaniem użytkownika mającym na celu uchwycenie jego intencji. W tradycyjnych interfejsach bazujących na komendach system oczekuje na wydanie komendy przez użytkownika, np. poprzez wskazanie jakiegoś elementu na ekranie (styl bezpośredniej manipulacji graficznej), naciśnięcie odpowiedniego klawisza (styl klawiszy funkcyjnych) lub wpisanie komendy (styl linii komend). System najczęściej generuje jakąś odpowiedź w postaci np. wyświetlanego komunikatu lub nadania sygnału dźwiękowego. W przypadku interfejsów wykorzystujących monitoring zachowań użytkownika istnieje niebezpieczeństwo błędnej interpretacji, dlatego wykorzystywany jest efekt multimodalności, tzn. jednoczesnej analizy więcej niż jednego źródła: głosu, ruchów oka [Tanaka 2005], gestów, ruchów ciała, rozmiaru źrenicy oka [Sobecki 2008], przewodności skóry, ciśnienia krwi, temperatury ciała, prozodii, szybkości oddechu, mimiki twarzy, składu chemicznego krwi, zapachu ciała, elektroencefalogramu mózgu

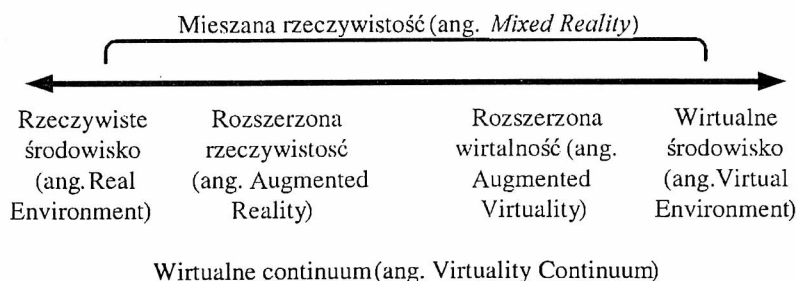
(EEG) lub nawet myśli. Czasami, dla niektórych użytkowników, zadowalające rezultaty można osiągnąć po długotrwałych treningach, które pozwalają interfejsom mózg-komputer (ang. *brain-computer interaction* – BCI) bazującym na detektorach EEG osiągać zadowalające rezultaty, np. 90% trafień w cel w czasie 10 sekund [Wickelgreen 2004]. Inną metodą przezwyciężenia problemów z wykorzystaniem interfejsów multimodalnych jest zastosowanie interfejsów typu znak + ograniczenie (ang. *token + constraints*) [Ullmer 2005], wymagają one zarówno operatorów, jak i operandów, które muszą współwystępować w celu utworzenia w pełni operowalnego interfejsu. W pracy [Ullmer 2005] analizowano wykorzystanie systemów z fizyczną reprezentacją zarówno znaków, jak i ograniczeń, które mogą być mechanicznie manipulowane.

### 2.1.1.5. Wirtualna rzeczywistość

Termin wirtualna rzeczywistość (ang. *Virtual Reality* – VR) był oryginalnie wykorzystywany do opisania immersyjnych doświadczeń sensorycznych ze sztucznie wytworzonymi światami [Chapman 2004]. Obecnie VR definiowana jest jako środek wizualizacji, manipulacji i interakcji człowieka z komputerem i złożonymi danymi. VR wykorzystuje pewien zestaw technologii, takich jak: hełm HMD (ang. *Head Mounted Display*), rękawice lub cały kombinezon interaktywny, urządzenia generujące dźwięk przestrzenny. Wirtualną rzeczywistość charakteryzują:

- intensywność (umożliwiająca zwrócenie szczególnej uwagi odbiorcy na wybranych obiektach),
- interaktywność (współdziałanie użytkownika z systemem poprzez interfejs),
- immersja (znaczný stopień absorbowania uwagi użytkownika dzięki trójwymiarowości bodźców),
- ilustracyjność (informacje powinny być podawane w sposób przejrzysty, opisowy i jasny),
- intuicyjność (informacje powinny być łatwo interpretowane).

W pracy [Miligram 94] zaprezentowano pojęcie *wirtualnego continuum* (ang. *Virtuality Continuum*) (rys. 1) odnoszącego się do pewnego zestawu klas obiektów, w którym środowiska rzeczywiste (ang. *Real Environment*) zajmują jeden jego koniec, a środowiska wirtualne (ang. *Virtual Environment*) drugi, przeciwny koniec. Oprócz czysto realnych lub wirtualnych środowisk w continuum wyróżniona została tzw. *mieszana rzeczywistość* (ang. *Mixed Reality* – MR), w skład której wchodzi: *rozszerzona rzeczywistość*



Rys. 1. Uproszczona wersja wirtualnego continuum [Miligram 94]

stość (ang. *Augmented Reality* – AR) i rozszerzona wirtualność (ang. *Augmented Virtuality* – AV). Dla przykładu, mianem AR określane są środowiska realne wzbogacone o pewne elementy wirtualne, np. w zastosowaniach architektonicznych rzeczywiste otoczenie określonych miejsc może zostać wzbogacone widokami dopiero planowanych budowli (budynków, dróg czy mostów). Z przypadkiem AR mamy do czynienia wówczas, gdy np. wirtualny świat gry 3D [Gawrysiak 1998] rozszerzony jest o pewne elementy rzeczywiste, np. postaci o wyglądzie lub choćby tylko twarzach ludzi.

### 2.1.2. Wybór stylu interakcji

Projektowanie, a zwłaszcza projektowanie interfejsów użytkownika, jest sztuką dokonywania wyboru [Spolsky 2001]. Wybór stylu interakcji jest jedną z głównych decyzji, które podejmuje projektant tworząc system interakcyjny [Newman 1996], są one również określone normami ISO 9421-10 [ISO 1996]. Także w przypadku systemów z rekomendacją interfejsu użytkownika wybór stylu interakcji jest istotą działania takiego systemu. Zaprezentowane style stanowią jedynie pewną gamę możliwych form interakcji użytkownika z systemem komputerowym, natomiast konkretna ich implementacja stanowi bardzo istotny element projektowania interfejsu użytkownika.

Pomocą w wyborze konkretnego stylu interakcji mogą być zasady projektowania zorientowanego na użytkownika (ang. *User-Centered Design* – UCD) przedstawione bliżej w punkcie 2.4. Użyteczność i dostępność systemów interakcyjnych. Dokonanie wyboru należy poprzedzić wypełnieniem następujących zaleceń [Katz-Haas 1998]: poznać użytkowników, przeanalizuj zadania i cele użytkowników oraz nie przesądzać o ostatecznych rozwiązaniach zbyt wcześnie. Mając zgromadzone informacje dotyczące użytkowników, zadań, które mają być realizowane oraz możliwych do zastosowania technologii, można przystąpić do dokonywania wyboru stylu interakcji.

Jedną z metod wyboru stylu interakcji jest zawężanie możliwego wyboru [Newman 1996]. W efekcie działania procesu zawężania można otrzymać w rezultacie jeden lub maksymalnie dwa style, które nadają się do zastosowania w danej sytuacji. Każdy ze stylów ma spore ograniczenia determinujące niejako możliwości jego wykorzystania. Przykładowo, jeżeli chcemy wybrać odpowiedni styl interakcji dla nowoczesnej zręcznościowej gry komputerowej (np. popularnej strzelaniny) [Gawrysiak 1998], to do wyboru mamy styl bezpośredniej manipulacji graficznej lub wirtualną rzeczywistość (VR) wraz z ewentualnym specjalistycznym manipulatorem z klawiszami funkcyjnymi. Oczywiście, tworząc grę dla tzw. salonu rozrywki, z możliwością zastosowania pewnych specjalistycznych instalacji, możemy zdecydować się na wybór VR z pewnymi elementami realnymi, tworząc system ze wzbogaconą VR (ang. *Augmented VR*), np. atrapą broni.

## 2.2. Hipertekst i hipermedia

Pojęcie hipertekstu zostało wprowadzone przez Nelsona w pracy [Nelson 1965], gdzie zdefiniowano jako „ciało pisanego lub graficznego materiału połączonego w tak złożony sposób, że nie może być konwencjonalnie prezentowane na papierze”. W innej defi-

nicji [Chapman 2004], hipertekst definiowany jest jako „tekst rozszerzony o odsyłacze, czyli wskaźniki innych porcji tekstu”. Definicje te odpowiadają np. stylowi pisania encyklopedii, w których pełno jest odesłań do szerszych objaśnień zamieszczonych w innych hasłach. Główną nowością hipertekstu jest nie tyle zastosowanie odsyłaczy, które można znaleźć już w bardzo starych, a nawet starożytnych pismach (np. odniesienia do pewnych cytatów z Biblii, które to odsyłacze bardzo wcześnie uzyskały swoją specyficzną formę), co mechanizm natychmiastowego zaprezentowania treści, na które wskazuje dany odsyłacz. Cechę tę oddaje w najlepszy sposób definicja Rady [Rada 1991], która określa hipertekst jako połączenie sieci semantycznej, tekstowej bazy danych oraz interfejsu umożliwiającego prezentację niniejszych treści.

Hipermedia, które to pojęcie również po raz pierwszy zdefiniował Nelson [Lennon 1997], można traktować jako logiczne rozszerzenie hipertekstu o obiekty multimedialne. Zdefiniowanie pojęcia multimediiów wymaga najpierw zdefiniowania pojęcia mediów. Tradycyjnie do mediów w dziedzinie systemów informacyjnych (SI) zaliczane są różne postaci prezentowanych w systemie informacji: tekst, obraz, dźwięk, animacje i wideo, chociaż w innych dziedzinach (np. naukach medioznawczych) termin ten może mieć wiele różnych znaczeń [Bednarek 2002]: środki przekazu (prasa, radio telewizja), sieci transmisyjne czy język naturalny. Multimedia cyfrowe są definiowane jako dowolna kombinacja dwóch lub więcej mediów w formie cyfrowej, zintegrowanych ze sobą w sposób wystarczający, aby mogły być prezentowane poprzez pojedynczy interfejs lub aby mogły być manipulowane przez pojedynczy program komputerowy [Chapman, 2004]. Interaktywność jest również traktowana jako „nowatorski wkład komputerów” do multimediiów [Chapman 2004], w niektórych definicjach [Katarzyniak 1995], oprócz wskazania na użycie wielu różnych mediów, wymaga się, aby aplikacja działała w trybie interakcyjnym, tzn. bezpośredniej kontroli użytkownika nad przebiegiem jej działania. Warunek ten najczęściej nie jest spełniony w przypadku większości prezentacji multimedialnych, realizowanych najczęściej za pomocą narzędzi i elementów multimedialnych takich samych jak interakcyjne aplikacje multimedialne. Dla rozróżnienia tego rodzaju realizacji w pracy Chapmana [Chapman 2004] wprowadzono terminy: *wielomedia* (ang. *multiple media*) i *multimedia*, w których kryterium rozróżniającym jest sposób percepcji użytkownika. W przypadku *wielomediów* użytkownik odbiera (ang. *perceive*) różne typy mediów (tzn. tekst, obrazy czy wideo) sekwencyjnie lub inaczej, przełącza się pomiędzy różnymi modalnościami; natomiast w przypadku *multimediiów* modalność odbioru mediów jest kombinowana, tzn. może być zarówno równoległa, jak i sekwencyjna [Chapman 2004].

Współcześnie za ojca idei hipertekstu uważa się Busha, dyrektora Biura Nauki i Rozwoju w administracji Roosevelta (prezydenta USA) [W3C 2003]. Bush w swojej pracy „Sposób, w jaki moglibyśmy myśleć” (ang. *As we may think*), przedstawił koncepcję systemu *memex*, który umożliwiłby w sposób zautomatyzowany (niekoniecznie na bazie systemu komputerowego) dodawanie odsyłaczy i adnotacji (zwanych śladami) w postaci stron z tekstem, tabel z wynikami badań czy nawet słownych komentarzy, które łączyłyby ze sobą różne dokumenty [Bush 1945].

W roku 1964 McLuhan w swojej pracy „Understanding Media: The Extensions of Man”, pisząc o popularnych mediach nadawczych (takich jak radio i telewizja) przedstawił wizję „globalnej wioski”, która uważana jest obecnie za podwaliny globalnej sieci jaką jest Internet [Lennon 1997].

Listę wizjonerów hipertekstu i jego wykorzystania w sieci globalnej zamyka wspomniany już twórca pojęć „hipertekst” i „hipermedia” – Nelson [Lennon 1997]. Był to jednak również realizator trwającego 25 lat projektu Xanadu, który obejmował implementację sieciowego systemu hipertekstowego. System ten, oprócz realizacji podstawowych funkcji, umożliwiał kontrolę wersji dokumentów, praw autorskich i pokrewnych, a także na księgowanie i wnoszenie opłat za wykorzystanie informacji. Z perspektywy lat widać, że rozwój i powodzenie wszelkich usług informacyjnych w sieci Internet było w pierwszej fazie ich działania zwykle oparte na darmowym i powszechnym dostępie, a dopiero po fazie popularyzacji określonej usługi można było wprowadzać opłaty i to najczęściej jedynie za wybrane elementy.

Twórcą pierwszego działającego systemu hipertekstowego NLS (ang. *oN-Line System*) był Engelbart [Lennon 1997]. Koncepcja tego systemu została przedstawiona w pracy [Engelbart 1962], w której projektowany system miał się składać z następujących artefaktów: komputery, zarządzany komputerowo zasób przechowywanych i udostępnianych informacji i urządzenia, które je prezentują. System NLS powstał w 1968 roku jako część większego projektu *Augment* i obejmował 100 000 fragmentów informacji, po których użytkownicy mogli nawigować podążając za odsyłaczami. Przy okazji warto również zaznaczyć, że Engelbart uważany jest za ojca takich urządzeń, aplikacji i usług, jak: mysz, przetwarzanie tekstu, system okien oraz e-mail.

W latach 70. XX w. tworzono kolejne systemy wykorzystujące ideę hipertekstu, jednak liczba tego typu projektów nie była zbyt duża. Kontynuowano oczywiście prace nad projektem Xanadu, ale także na bazie powstających w tym czasie platform systemowo-sprzętowych opartych na mini i mikroprocesorach tworzono systemy KMS (*Knowledge Management System*) i *Dynabook* [Lennon 1997].

Lata 80. ubiełego wieku to dynamiczny rozwój mikrokomputerów oraz ich zastosowania w szkołach, kiedy przede wszystkim dla Mackintosha, jak również i dla innych platform opracowano wiele narzędzi, takich jak *Hypercard*, *Hypernotes*, *NoteCards* czy *Toolbook*. Narzędzia te umożliwiały specjalistom z określonych dziedzin wiedzy tworzenie, hipertekstowych lub hipermedialnych systemów informacyjnych wykorzystywanych najczęściej w celach edukacyjnych lub naukowych [Steinbrink 1993]. Pozwalały one bowiem na zebranie, uporządkowanie i udostępnienie potencjalnemu odbiorcy informacji w sposób o wiele bardziej wygodny niż dostarczenie takich samych materiałów w postaci źródłowej, kolekcji niepowiązanych dokumentów w postaci cyfrowej lub tworzonych metodami analogowymi multimedialnych systemów wspomagających edukację [Strykowski 1984].

Na bazie tych idei także w Polsce pod koniec lat 80. XX w. w Zakładzie Systemów Informacyjnych Politechniki Wrocławskiej realizowane były projekty, w ramach których powstawały hipertekstowe systemy informacyjne. Przeznaczone one były początkowo na pojedyncze stanowiska, a z czasem powstały wersje sieciowe. Były to systemy



*Verte!* i opracowany m.in. przez Autora SWIP [Choroś 1992]. Obydwa te systemy zawierały kompendia wiedzy z zakresu kompresji obrazów [Skarbek 1998], przy czym w drugim systemie wykorzystywano specyficzną reprezentację wiedzy w postaci tzw. problemów naukowych [Sobecki 1995]. Mimo że projekty te i ich rezultaty nie były powszechnie znane, niewątpliwie przyczyniły się do w miarę szybkiego zasymilowania WWW przez uczestników tych projektów, co z kolei zaowocowało m.in. zaimplementowaniem pierwszego w Polsce dostępnego pod WWW od 1994 r. katalogu książek Biblioteki Głównej Politechniki Wrocławskiej [Kołcio 1995] na podstawie koncepcji opracowanej przez Autora.

## 2.3. Technologie webowe

W momencie swojego powstania usługa WWW opierała się na języku HTML (ang. *HyperText Markup Language*) służącym do tworzenia stron hipertekstowych, protokole HTTP (ang. *Hypertext Transfer Protocol*) oraz adresowaniu URL (ang. *Uniform Resource Locator*, dawniej *Universal Resource Locator*).

Pierwsza wersja HTML datowana jest na rok 1992. Wcześniej, w CERN, prowadzono prace nad udostępnianiem informacji hipertekstowej z użyciem technologii NeXT [Feizabadi 1998] rozwijanej w owym czasie przez Steva Jobbsa – współzałożyciela firmy Apple. HTML jest językiem formatowania dokumentów bazującym na standardzie SMGL (ang. *Standard Generalized Markup Language*), który jest określony przez Międzynarodowy Standard ISO 08879:1986. Język HTML umożliwia tworzenie w prosty sposób dokumentów hipermedialnych składających się ze sformatowanego tekstu, odsyłaczy hipertekstowych i grafiki, oraz umożliwia ich wyświetlenie na różnych platformach sprzętowych i programowych [Sobecki 1996]. Język HTML był rozwijany przez ponad 10 lat, następnie podjęto prace nad językiem XHTML (ang. *Extensible HyperText Markup Language*). XHTML jest rodziną typów dokumentów, które reprodukuje rozszerzony HTML sformułowany w XML (ang. *Extensible Markup Language*). Oznacza to, że XHTML bazuje na XML i przeznaczony jest do prezentacji w przeglądarkach zaprojektowanych do pracy z dokumentami XML. Od 2000 roku W3C (ang. *World Wide Web Consortium*) rekomenduje następcę HTML, a mianowicie język XHTML.

Od kilku lat w różnego rodzaju aplikacjach webowych coraz większą rolę ma zastosowanie rodziny języków XML [W3C 1996]. Język XML doskonale nadaje się do wymiany danych pomiędzy różnymi aplikacjami i platformami programowymi i sprzętowymi. Poprawność dokumentów XML weryfikowana może być przez zastosowanie DTD (ang. *Document Type Definition*) lub – XML Schema. Język XML oraz cała rodzina języków opartych na XML ma również wiele innych zastosowań.

HTTP jest protokołem sieciowym warstwy aplikacji [Berners-Lee 1996]. Zadaniem HTTP jest zapewnienie komunikacji pomiędzy procesem wywołującym (realizowanym przez klienta) a bezstanowym procesem obsługującym (realizowanym przez serwer). Połączenie klienta z serwerem, np. w celu pobrania jakiegoś pliku (transakcja), prze-



biega w czterech etapach: otwarcie połączenia, żądanie klienta, odpowiedź serwera i zakończenie połączenia.

Pierwszą wersją HTTP zaimplementowaną przez Bernersa-Lee była wersja oznaczona jako 0.9. Protokół ten obsługiwał tylko jedną metodę komunikacji, narzucał ograniczenia na typ zawartości, nie zwracał informacji o zawartości oraz nie dawał możliwości klientowi przesyłania informacji do serwera. Protokół HTTP 1.0 nie miał już niedoskonałości swojego poprzednika, a także wykorzystywał nagłówki typu MIME i umożliwiał serwerowi odpowiadanie za pomocą kodów statusów. Podstawową wadą HTTP 1.0 była konieczność wielokrotnego otwierania i zamykania połączenia, gdy do wyświetlenia pojedynczego dokumentu HTML należało wczytać inne pliki, np. obrazy lub podłączone dokumenty. Powodowało to zwielokrotnienie liczby połączeń i stratę czasu na otwieranie i zamykanie połączenia. Problemy te usunęło wprowadzenie protokołu w wersji 1.1, w którym wszystkie żądania realizowane są w pojedynczym połączeniu.

URL (ang. *Uniform Resource Locator*) jest ostatnim z elementów podstawowych technologii WWW. URL jest adresem fizycznym obiektów, które można wyszukać, używając dostępnych w Internecie protokołów. Kompletny URL składa się z dwóch części: specyfikatora schematu oraz ciągu znaków o formacie zgodnym ze schematem tego specyfikatora. Dla WWW specyfikatorem jest http. Ciąg znaków zaczyna się od dwóch znaków "http://", po których następują kolejno: adres serwera, port, ścieżka oraz ewentualnie kotwica i indeksy.

Podstawowymi typami transakcji HTTP są: GET i POST [Berners-Lee 1996]. W przypadku gdy przeglądarka wysyła żądanie pobrania określonej strony, HTML realizuje to za pomocą transakcji GET HTTP. Transakcja ta umożliwia również przesłanie pewnego ciągu znaków reprezentującego zapytanie wraz z adresem URL identyfikującym program CGI (ang. *Common Gateway Interface*), który może odpowiednio przetworzyć te dane i zwrócić odpowiedź w odpowiedniej formie. Ten typ transakcji był wykorzystywany we wspomnianej już w punkcie 2.2 aplikacji webowej udostępniającej zasoby biblioteczne Politechniki Wrocławskiej [Kołcio 1995]. Drugi typ transakcji – POST – nie wykorzystuje ciągu znaków reprezentującego zapytanie w postaci nagłówka żądania przeglądarki internetowej, tak jak w przypadku transakcji GET, a zamiast tego dane do programu CGI przesyłane są w sekcji zawartości danych tego zapytania. Bardzo często wykorzystywaną metodą jest wprowadzenie odpowiednich danych do formularza zdefiniowanego na stronie HTML, a następnie przesłanie ich za pomocą transakcji POST do odpowiedniego programu rezydującego na serwerze [Knucles 2005]. Transakcja POST została wykorzystana w innej aplikacji webowej, która udostępniała zasoby katalogu bibliotecznego Politechniki Wrocławskiej i kilku innych bibliotek z wykorzystaniem serwisu WAIS [Jakubowski 1997] i [Katarzyniak 1997], oraz w systemie upowszechniającym dorobek naukowy pracowników Politechniki Wrocławskiej [Sobecki 1997].

Standardowe strony HTML są statyczne, tzn. pobierając daną stronę otrzymamy zawsze taką samą zawartość. Atrakcyjność systemów webowych polega na możliwości wpływania użytkownika na zawartość oraz postać otrzymywanych informacji.

Zwykle systemy webowe działają w technologii klient–serwer. Organizacja przetwarzania polega na takim rozdzieleniu przetwarzania, że klient odpowiada za zadanie „pytania” do serwera, a ten na podstawie zasobów stara się na nie odpowiedzieć. Prezentacją tej odpowiedzi zajmuje się ponownie klient. Bardzo często wyróżnia się dwa typy klientów: tzw. „cienkiego” i „grubego”. Cienki klient wykorzystuje jedynie przeglądarkę do tworzenia instancji interfejsu, natomiast gruby umożliwia również wykonywanie innych aplikacji.

Kiedy klient łączy się bezpośrednio z bazą danych lub monolitycznym serwerem aplikacji, wówczas architektura aplikacji jest dwuwarstwowa. Ostatnio coraz częściej cienki klient, który nie obejmuje logiki biznesowej, a jedynie elementy interfejsu użytkownika, łączy się z serwerem aplikacji, który implementuje logikę biznesową, a następnie komunikuje się z serwerem bazy danych, który przechowuje „surowe” dane wykorzystywane przez aplikację. Architektura tego typu nazywa się trzywarstwową i jest specjalnym przykładem architektury  $n$ -warstwowej.

Obecnie projektanci mają do dyspozycji wiele różnorodnych technologii programowych umożliwiających na zdynamizowanie projektowanych, a następnie implementowanych serwisów webowych. Technologie programowania systemów webowych można przedstawić w zależności od warstwy, w której są one zaimplementowane, tzn.: warstwa klienta (prezentacji), warstwa logiki biznesowej i warstwa bazy danych. Ich ewolucyjny rozwój charakteryzują następujące etapy: HTML, DHTML, Aplikacje internetowe i RIA oznaczają odpowiednio:

- HTML: umożliwia prezentację jedynie statycznej zawartości,
- DHTML: oznacza dynamiczny (ang. *Dynamic*) HTML, tzn. HTML rozszerzony o możliwości wykorzystania skryptów JavaScript, stylów kaskadowych (ang. *Cascading Style Sheets* – CSS) oraz obiektowy model dokumentu (ang. *Domain Object Model* – DOM),
- Webapp: aplikacja webowa (ang. *Web Application*) jest aplikacją udostępnianą przez sieć Internetu lub intranetu, co umożliwia jej wykorzystanie bez instalowania dodatkowego oprogramowania,
- RIA: oznacza wzbogaconą aplikację internetową (ang. *Rich Internet Application*), kategoria zostanie dokładniej zaprezentowana podczas omawiania platformy Flash w punkcie 2.3.1.1.

### 2.3.1. Technologie warstwy prezentacji

Warstwa klienta bazuje na przeglądarce internetowej. Obecnie najpopularniejszymi przeglądarkami są odpowiednio: Microsoft Internet Explorer, Mozilla FireFox, Netscape Navigator i Opera. Przeglądarki internetowe komunikują się z serwerami WWW wykorzystując protokół HTTP lub np. HTTPS i wyświetlają na komputerze klienta dokumenty utworzone z wykorzystaniem języka HTML, XHTML, XML lub CCS.

W celu zdynamizowaniu prezentowanych treści, statyczne dokumenty mogą zostać uzupełnione o elementy JavaScript (JS), które pozwalają na zmianę stanu strony webowej po jej „zrenderowaniu” przez oprogramowanie przeglądarki. JavaScript to obiektowy

wo zorientowany język skryptowy opracowany przez firmę Netscape. Połączenie technologii HTML, XHTML i CCS z JavaScript nazywane jest DHTML, i charakteryzuje się zarówno walidacją, jak i wykonaniem kodu JavaScript po stronie klienta [Knuckles 2005].

Pokróćce zostaną przedstawione wybrane technologie warstwy prezentacji, które mają istotne znaczenie w tworzeniu multimedialnych interfejsów systemów webowych RIA, oraz mechanizm HTTP cookies i technologię DoubleClick, które można wykorzystać w personalizacji interfejsów i rekomendacji.

### 2.3.1.1. Aplikacje RIA

Jednym z najbardziej popularnych narzędziem do tworzenia aplikacji RIA jest Adobe Flash (uprzednio zwany Macromedia Flash) [Allaire 2002]. Flash jest narzędziem autorskim do tworzenia aplikacji webowych i off-line, wykorzystującym do ich tworzenia koncepcję filmu [Underhal 2002]. Oznacza to, że w aplikacji wydarzenia umieszczane są na osi czasu i pojawiają się na scenie klatka po klatce. Flash jest narzędziem na tyle atrakcyjnym, że obecnie trzecia firma dba o jego rozwój [Waldron 2006]. Były to FutureWave Software do roku 1996, Macromedia do roku 2005 oraz od tego czasu firma Adobe. Firma Adobe wprowadziła od 2005 roku jedną wersję Adobe Flash pod nazwą Flash CS3 Professional. Oprócz kompletnej implementacji języka Action Script 3.0, którego wersja 1.0 pojawiła się po raz pierwszy w Macromedia Flash 5.0 [Underhal 2002a], ma lepszą integrację z innymi „flagowymi” produktami Adobe – Photoshop i Illustrator. Przydatność tego narzędzia do tworzenia spersonalizowanych interfejsów użytkownika w webowych systemach rekomendacyjnych została potwierdzona poprzez zastosowanie w następujących implementacjach:

- system rekomendujący strategię nauczania *Prawo Jazdy* [Sobecki 2003b],
- system informacyjny *Megane* [Sobecki 2004b],
- system rekomendujący filmy *The Movie* [Sobecki 2006],
- system rekomendujący przepisy kulinarne *Kucharek* [Sobecki 2006a].

Aplikacje RIA mogą być również tworzone z użyciem innych technologii; należą do nich m.in.: Java, AJAX i XUL. Spośród nich najdłuższą historię i największą popularność ma technologia Java firmy Sun Microsystems. Wykorzystanie języka Java w sieci WWW zaczęło się od stworzenia w Javie przeglądarki internetowej WebRunner („wcześnie wcielenie HotJava” [O’Connell 1995]). Następnie pojawił się pomysł osadzanych na stronie WWW apletów Javy, które mogły być uruchamiane z wykorzystaniem dowolnej przeglądarki internetowej bez konieczności wcześniejszego ich instalowania oraz dbania o odpowiednią wersję. Dzięki takim założeniom aplety Java mogły być uruchamiane na dowolnych platformach sprzętowych i systemowych pod warunkiem posiadania przeglądarki i wirtualnej maszyny Java (ang. *Java Virtual Machine*). Jedną z wielu zalet apletów Java odróżniającą je od aplikacji desktopowych, także napisanych w środowisku Java, jest ich odseparowanie od lokalnych zasobów komputera, na którym jest uruchamiany. W ten sposób zwiększono bezpieczeństwo kosztem pewnych funkcjonalności, na przykład kosztem braku dostępu do plików na komputerze użytkownika. Jedną z łatwo zauważalnych różnic pomiędzy Flashem i Javą jest stosowanie grafiki we-

ktorowej w pierwszym i rastrowej w drugim przypadku. Aplikacje w Javie, podobnie jak we Flashu, mogą mieć pewne wady, do których należy zbyt duży ich rozmiar, co powoduje, że za każdym razem określona porcja danych, sięgająca nawet dziesiątków megabajtów, musi być przesłana przez sieć. Przez wiele lat Java była rozwijana przez firmę Sun Microsystems, aż do ogłoszenia w 2006 r. otwarcia jej kodu. Kod Javy został opublikowany 8 maja 2007 r. pod generalną licencją publiczną (ang. *General Public License* – GPL).

Kolejną technologią umożliwiającą tworzenie interaktywnych aplikacji webowych jest AJAX (ang. *Asynchronous JavaScript and XML*). Sama nazwa sugeruje połączenie technologii JavaScript i XML-a, natomiast asynchroniczność odnosi się do faktu niezależności ładowania zwykłej strony webowej od jej dodatkowych komponentów. Jednym z popularniejszych narzędzi do tworzenia aplikacji w technologii AJAX jest Google Web Toolkit (GWT), do najpopularniejszych zaś aplikacji stworzonych w tej technologii należą: Google Maps i Gmail oraz Google Docs & Spreadsheet [Google 2007]. AJAX jest połączeniem kilku technologii [Garret 2005]: XHTML i CSS, DOM, XML i XSLT, XMLHttpRequest i JavaScript.

Ostatnią omawianą tu technologią umożliwiającą zdynamizowanie interfejsów jest XUL (ang. *XML User-Interface Language*) [XUL 2007]. Jest to język opracowany przez Mozillę i możliwy do uruchomienia pod przeglądarkami opartymi na silniku GECKO, a mianowicie: Firefox, Thunderbird, ChatZilla i Netscape Nawigator, bez względu na system operacyjny lub platformę sprzętową, na której działa dana przeglądarka. Do podstawowych cech i zalet języka XUL należą [XUL 2007]: „silny” język znacznikowy, bazowanie na standardach (XML, CSS, JavaScript i DOM), przenoszalność międzyplatformowa, separacja warstwy prezentacji od logiki aplikacji oraz łatwość przystosowania, lokalizacji i brandowania. Niestety, mimo wielu zalet, technologia ta nie zyskała zbyt wielkiej popularności, a liczba aplikacji, szczególnie w porównaniu do aplikacji Flasha, jest niewielka.

Technologia RIA, pomimo iż nastęrcza programistom sporych trudności w łączeniu komponentów różnego typu oraz wymaga zastosowania specyficznych metod do określenia ich użyteczności [Loosley 2006], jest coraz częściej stosowana. Już obecnie aplikacja Google Maps ma ogromną rzeszę globalnych użytkowników. Można przypuszczać, że aplikacje Google Docs & Spreadsheet, które są darmowe, niezależne od platformy oraz nie wymagają instalacji mogą zyskać również sporą liczbę użytkowników, w perspektywie przekraczającą liczbę użytkowników aplikacji biurowych, takich jak Microsoft Office.

#### 2.3.1.2. HTTP Cookie i DoubleClick

Systemy webowe wykorzystują do komunikacji z użytkownikiem bezstanowy protokół HTTP. Każde żądanie klienta (tj. generowane przez przeglądarkę internetową użytkownika) zawiera obok adresu URL żądanego dokumentu, także jego adres IP, wersję wykorzystywanego protokołu oraz rodzaj i wersję wykorzystywanej przeglądarki internetowej. Już tak skąpe dane, szczególnie skojarzone z odpowiednimi danymi o użytkowniku mogą być bardzo cennym źródłem informacji przydatnym w modelowaniu użytkownika [Berners-Lee 1996].

Specyfika sieci Internet, bardzo popularne wykorzystywanie łączy niestałych, dynamicznych adresów IP, serwerów Proxy, tych samych stacji roboczych przez wielu użytkowników, np. w kawiarenkach Internetowych czy pracowniach komputerowych, powodują, że nazbyt często nie da się jednoznacznie przypisać konkretnemu użytkownikowi określony adres IP. Dlatego też opracowano inne metody umożliwiające taką identyfikację. Przede wszystkim należy tu wymienić mechanizm HTTP *cookies* lub *cookie* (ciasteczko), zwane również *Web cookies* lub *tracking cookies*, który został opracowany przez firmę Netscape Communications i wprowadzony w przeglądarce Netscape Nawigator version 2 [Knuckles 2005] i opisanej w specyfikacji RFC 2109 [Kristol 1997].

*Cookies* wykorzystywane są do identyfikowania użytkowników oraz zapisywania danych wprowadzonych w różnego rodzaju formularzach, a także zawierają dane o odwiedzanych stronach oraz prezentowanych użytkownikowi reklamach [Whalen 2002]. *Cookies* są zapisami generowanymi przez serwer i dokonywanymi przez przeglądarkę internetową po stronie klienta na podstawie odpowiednich poleceń umieszczonych w kodzie dokumentów webowych. Pozwalają one na ustawienie i następnie pobranie odpowiednich wartości, do których należą [Whalen 2002]: nazwa *cookie*, wartość *cookie*, data wygaśnięcia, ścieżka dostępu i domena posiadacza *cookie*, informacja o wykorzystaniu SSL. Każde *cookie* może być odczytywane jedynie przez strony pochodzące ze wskazanej ścieżki i domeny określonych w parametrach *cookie*.

Mechanizm *cookies* działa dwufazowo. W pierwszej fazie są one zapisywane na komputerze użytkownika, w drugiej są one przesłane z komputera użytkownika do uprawnionego serwera systemu webowego. Zapisy dokonywane są bez powiadamiania użytkownika o tym fakcie. Przeglądarki internetowe mają oczywiście opcje umożliwiające wyłączenie tego mechanizmu lub zapisywanie *cookies* jedynie w plikach tymczasowych, jednak z drugiej strony wiele serwisów koniecznie wymaga jego włączenia, np. Onet. Wyłączenie *cookies* nie zapobiega przesyłania zapisów dokonanych przed wyłączeniem. W tym celu należy usunąć wszystkie pliki zawierające *cookies*. Niektóre systemy webowe, np. bankowości elektronicznej, dla bezpieczeństwa polecają, aby podczas korzystania z ich usług mechanizm *cookies* był wyłączony.

Jak wynika z właściwości *cookies* tylko serwis, który zapisał *cookie* ma uprawnienia do jego odczytania. Jednak, jeżeli użytkownik, który nie wyłączył tego mechanizmu w odpowiedniej kartotece, może zobaczyć zapisy dokonane przez serwisy, których jak mu się jedynie wydaje, nigdy nie odwiedzał. W tym miejscu należy wrócić do istoty protokołu HTTP i dokumentów HTML. W przypadku gdy dana strona zawiera oprócz tekstu jakieś obrazy, wykorzystuje ramy (ang. *frames*) lub zewnętrzne style kaskadowe CCS, wówczas konieczne jest wywołanie więcej niż jednej sesji HTTP. Tak doczytywane elementy nie muszą wcale znajdować się na tym samym serwerze, o tej samej domenie, ale mogą pochodzić od organizacji specjalizujących się w zakresie badania zachowań użytkowników, należy do nich DoubleClick, który stosuje technologię chronioną patentem [Whalen 2002].

Wstawiane do dokumentu elementy to najczęściej niewidoczne obrazki wielkości 1×1 pikseli. Takie wstawki zwane są czasem sieciowymi pluskwami (ang. *web bugs*) [Smith 2004]. Mechanizm ten działa następująco [Whalen 2002]:

- dysponent pewnego serwisu współpracujący np. z Doubleclick umieszcza na swoich stronach odpowiednie pluskwy sieciowe,
- po połączeniu się użytkownika z taką stroną aktywowana jest pluskwa sieciowa, która najpierw sprawdza czy użytkownik ma już plik *cookie* Doubleclick i jeżeli nie, zostaje mu nadany unikatowy numer identyfikacyjny i zapisany w pliku *cookie*, w przeciwnym razie następuje przesłanie do serwera Doubleclick zawartości *cookie*, a następnie ustawieniu nowych,
- odpowiednie dane dotyczące użytkownika mające wpływ na odpowiednie zakwalifikowanie tego konkretnego użytkownika wymieniane są między serwisem webowym a DoubleClick.

Zastosowanie pluskiew sieciowych na stronach WWW może dostarczyć wiele różnorodnych i bardzo cennych danych dotyczących zachowań i cech użytkownika. Należą do nich m.in. następujące dane [Whalen 2002]:

- dane dotyczące liczby odwiedzin określonej strony,
- dane dotyczące danych demograficznych użytkownika,
- dane dotyczące adresów odwiedzanych stron,
- dane dotyczące prezentowanych użytkownikowi reklam, tzn. ich rodzaju i liczby pojawień,
- dane umożliwiające skojarzenie reklamy z faktem zakupu reklamowanego towaru lub przynajmniej odwiedzenia przez użytkownika odpowiednich stron z informacjami o reklamowanym towarze, a przez to pozwalające na mierzenia efektywności rekomendacji,
- dane zawierające zapytania użytkownika kierowane do wyszukiwarek internetowych,
- raporty o wersji i ustawieniach przeglądarki internetowej,
- wyniki synchronizacji *cookie*, polegającej na wzajemnej wymianie informacji o użytkowniku pomiędzy różnymi firmami,
- wyniki sprawdzania liczby otwarć danego banera reklamowego.

Jak widać wykorzystanie pluskiew sieciowych daje bardzo duże możliwości w zdobywaniu informacji o użytkownikach, które mogą być wykorzystane do budowy modelu użytkownika w systemie interakcyjnym, jednak ich wykorzystanie, choć wydaje się być legalne, często jest oceniane jako niezbyt etyczne [Whalen 2002] lub przynajmniej kontrowersyjne. Należy jednak przyznać, że z reguły firmy, takie jak DoubleClick umożliwiają deindywidualizację zapisów przez zamianę unikatowego numeru identyfikacyjnego na numer jednakowy dla wszystkich chętnych do zachowania swojej anonimowości, jednak liczba takich użytkowników jest niewielka.

### 2.3.2. Technologie warstwy aplikacji

W dwuwarstwowej lub trzywarstwowej aplikacji webowej obok warstwy prezentacyjnej, której możliwe do wykorzystania technologie przedstawiono powyżej, występuje przede wszystkim warstwa aplikacji. Przez kilkanaście lat rozwoju aplikacji webowych powstało już bardzo wiele różnorodnych technologii.



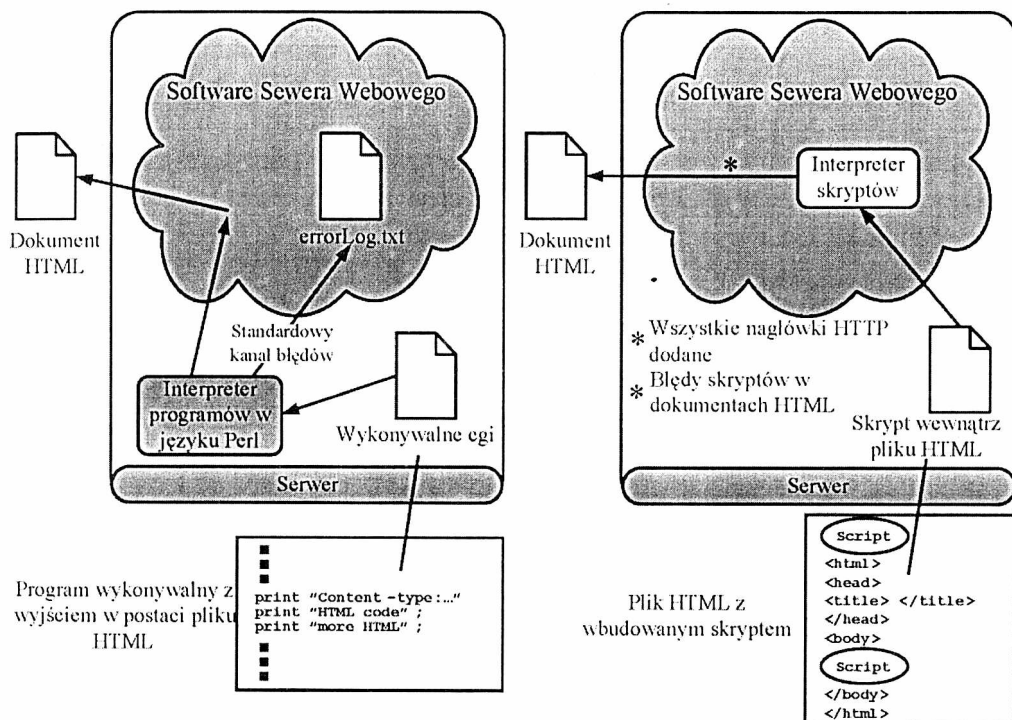
Aplikacje webowe różnią się od innych aplikacji komputerowych przede wszystkim tym, że nie muszą być instalowane na lokalnym komputerze, a w zamian uruchamiane są poprzez wywołanie przez przeglądarkę internetową określonej strony [Knuckles 2005]. Aplikacje te mogą być tworzone z wykorzystaniem różnorodnych technologii [Sobecki 2004b], [Sobecki 2006].

Najstarszą technologią wykorzystywaną do tworzenia aplikacji webowych jest CGI (ang. *Common Gateway Interface*) [CGI 1995], która po raz pierwsza została zaimplementowana na serwerze NCSA HTTPd serwer. Technologia ta umożliwia uruchamianie programów napisanych w dowolnym języku, byleby ich kod uruchomieniowy był wykonywalny na serwerze [CGI 1995]: C/C++, Fortran, PERL, TCL, Any Unix Shell, Visual Basic, AppleScript. Jak widać, technologia ta umożliwia również wykorzystanie języków skryptowych, które nie wymagają kompilacji, np. skryptowy język PERL był swego czasu najczęściej wykorzystywany do tworzenia programów CGI. Oczywiście inne serwery WWW przeznaczone na różne systemy umożliwiały również uruchamianie innych programów, np. plików BAT na systemach firmy Microsoft [Kołcio 1995]. Jednym z mankamentów technologii CGI jest konieczność każdorazowego uruchomienia programu po otrzymaniu zapytania od użytkownika. W przypadku gdy program ten ma sporą wielkość lub jego wywołanie następuje bardzo często, powoduje to zbędne obciążenie systemu [Knuckles 2005].

Technologia CGI charakteryzuje się m.in. przechowywaniem pewnych wartości związanych z każdą transakcją w zmiennych środowiskowych, do których należą m.in. nazwa domeny serwera, ciąg zapytania dołączony do URL, adres IP klienta oraz metoda transakcji (GET lub POST) [Knuckles 2005]. W przytoczonej już implementacji aplikacji webowej udostępniającej zasoby katalogu bibliotecznego Biblioteki Głównej i OINT Politechniki Wrocławskiej [Kołcio 1995] użytkownik wypełniał formularz zamieszczony na stronie HTML, a następnie po wybraniu opcji „wyszukaj” wprowadzone dane zostały wysyłane metodą GET do serwera wraz ze wskazaniem programu BAT, który zapisywał wprowadzone dane do pewnego pliku. Plik ten, dostępny w sieci Novell, był czytany przez specjalny program, który przekazywał dane do systemu APIN-SZUK. System APIN-SZUK był elementem systemu bibliotecznego działającego w sieci Novell, a jego zadaniem było przeszukiwanie katalogu książek. Program ten zapisywał wyniki wyszukiwania w swoim formacie, z którego następnie były formatowane do HTML i zapisywane jako plik o odpowiedniej nazwie i miejscu lokalizacji. Następnie plik ten za pomocą mechanizmu CGI był przekazywany do użytkownika jako odpowiedź na zapytanie. Jako, że zastosowane komputery wykorzystywały systemy, które praktycznie nie były wielozadaniowe, w cały proces były zaangażowane trzy serwery: jeden serwer WWW, jeden formatujący i jeden serwer Novell z systemem APIN-SZUK [Kołcio 1995]. Należy tu dodać, że był to absolutnie pierwszy katalog biblieczny w Polsce udostępniany przez WWW.

Ze względu na konieczność uruchamiania po każdym zapytaniu wskazanego w zapytaniu programu zostały opracowane inne technologie, takie jak: PHP, ASP, JSP i ColdFusion. Różnice pomiędzy różnymi preprocesorami (np. PHP i ASP) a mechanizmem CGI przedstawiono na rysunku 2. PHP jest również nazwą języka skryptowego





Rys. 2. Preprocesory hipertekstowe a programy CGI [Knuckles 2005]

uruchamianego po stronie serwera [Helling 2003]. Jako rozwiązanie otwarte istnieje możliwość jego zainstalowania na serwerach WWW Open Source, takich jak Apache [Sobecki 2004b] oraz serwerach komercyjnych, takich jak IIS (ang. *Internet Information Services*). Kod PHP jest osadzony na stronie HTML i gdy występuje żądanie pobrania określonej strony zawarty w niej skrypt jest wykonywany na serwerze WWW, a wynik, najczęściej w postaci kodu HTML, przesyłany jest do klienta realizującego żądanie pobrania określonej strony. Historia technologii PHP jest prawie tak samo długa jak CGI, gdyż został on opracowany przez Lerforda w 1994 i jak wiele technologii internetowych jest produktem Open Source. Język PHP jest z tego powodu bardzo popularny, według stanu z kwietnia 2007 r. 20 016 421 domen i 1 208 663 adresów IP używało technologii PHP [PHP 2007].

Technologię PHP wykorzystano np. w implementacji systemu informacyjnego w promocji produktów z adaptacyjnym interfejsem *Megane* [Sobecki 2004b], gdzie została ona zastosowana jako połączenie pomiędzy warstwą prezentacji i bazą danych. Skrypty PHP realizują w tym systemie tzw. inteligencję biznesową, czyli warstwę aplikacji. W warstwie tej realizowane są m.in. następujące funkcje: rejestracja użytkowników w systemie, logowanie użytkowników, aktualizacja profilu użytkownika, personalizacja interfejsu, grupowanie użytkowników oraz wyznaczanie reprezentanta interfejsu dla każdej z grup metodą konsensusu.

Obie poprzednio przedstawione technologie są darmowe i otwarte. Kolejna technologia nosi nazwę ASP (ang. *Active Server Pages*) i została wprowadzona na rynek przez firmę Microsoft w 1996 roku [Knuckles 2005]. Technologia ASP dostępna na serwery WWW tej firmy, takie jak IIS i PWS (ang. *Personal Web Server*) [Witkowski 2002]. Skrypty ASP mogą być programowane w różnych językach, takich jak: Visual Basic Script lub JavaScript, które domyślnie są zainstalowane na serwerach IIS i PWS. Od 2002 roku firma Microsoft rozwija nowszą wersję tej technologii, a mianowicie ASP.NET, która obecnie dostępna jest w wersji 3.0 [ASP 2007]. Ze względu na zintegrowanie tej technologii z systemem operacyjnym ma ona możliwość zwiększenia funkcjonalności dzięki zastosowaniu bibliotek DLL oraz możliwości: autooptymalizacji kodu (*ASP Self-Tuning*), kodowania skryptów ASP do postaci binarnej oraz użycie Windows Scripts Components [Witkowski 2002].

Z wykorzystaniem technologii ASP i ASP.NET zostało zaimplementowanych kilka aplikacji rekomendacyjnych: *Kucharek* [Sobecki 2006a], *The Movies* [Sobecki 2006] oraz *The Movie* [Nguyen 2007].

Opócz wymienionych technologii warstwy aplikacji CGI, PHP i ASP, wykorzystywane są również inne technologie, takie jak: JSP (Java Server Pages), Pike i ColdFusion. Najpopularniejszą jest pierwsza z nich, oparta na języku Java i podobnie jak ten język programowania została opracowana przez firmę Sun [Witkowski 2002]. Technologia ta wymaga zainstalowania na serwerze WWW serwera serwetów bądź serwera aplikacji. Strony JSP oprócz kodu HTML zawierają kod napisany w języku Java, który może wykorzystywać komponenty JavaBeans. W przypadku pierwszego wywołania określonej strony osadzony na niej kod jest kompilowany do postaci binarnej, która jest pobierana z pamięci przy kolejnych wywołaniach strony. Technologia JSP jest szczególnie często wykorzystywana w aplikacjach mobilnych, z tego powodu, że na platformach klienc-kich język Java jest podstawowym językiem oprogramowania aplikacji, również webowych.

### 2.3.3. Technologie warstwy bazy danych

Baza danych jest koniecznym elementem w trójwarstwowej technologii webowej. W liczącej już blisko 50 lat historii rozwoju baz danych zostało stworzonych wiele ich modeli: hierarchiczny, sieciowy i relacyjny [Date 1981]. Ich wykorzystanie, a szczególnie najpowszechniej używanych obecnie relacyjnych systemów zarządzania bazami danych – RDBMS (ang. *Relational Database Management System*), pozwalana m.in. na [Helling 2003]: szybszy dostęp do danych w porównaniu plikami jednorodnymi oraz formułowanie zapytań o dane spełniające określone kryteria. Ponadto RDBMS mają mechanizmy umożliwiające równoczesny dostęp do danych, umożliwiają swobodny dostęp do danych oraz posiadają wbudowany system przywilejów. Oprócz RDBMS coraz większą popularność zyskują obiektowe bazy danych ODBMS (ang. *Object Database Management System*) [Atkinson 1989].

W aplikacjach webowych stosowane są różnorodne systemy zarządzania bazą danych zarówno darmowe, jak i komercyjne. Wśród najpopularniejszych RDBMS można

wymienić systemy darmowe: MySQL [Welling 2003] i PostgreSQL oraz komercyjne: Oracle, MS SQL Server, InterBase oraz FireBird. Natomiast wśród najpopularniejszych ODBMS można wymienić [Riccardi 2001]: Jasmine, Gemstone, O2, Object Store, Objectivity/DB, Versant ODMBS.

Rozwiązania darmowe i Open Source bardzo często dostarczają wystarczająco bogatą funkcjonalność, która pozwala z dużą swobodą implementować różnorodne systemy rekomendacyjne. Wspomniany już system *Megane* wykorzystywał bazę danych MySQL [Sobecki 2004b], natomiast system rekomendujący przepisy kulinarne *Kucharek* [Sobecki 2006] i system rekomendujący filmy w wypożyczalni *Movie* [Rakowski 2006] wykorzystywały system MS SQL.

## 2.4. Użyteczność i dostępność systemów interakcyjnych

Pojęcie użyteczności (ang. *usability*) ma wiele definicji, np. według normy PN-EN ISO 9241-11 użyteczność definiowana jest jako: *stopień, do którego produkt może być używany przez określonych użytkowników do osiągnięcia określonych celów uzyskując: skuteczność, wydajność i satysfakcję w określonym kontekście użycia*. Natomiast wg normy ISO/IEC 9126-1 użyteczność jest elementem Modelu dla Zewnętrznej i Wewnętrznej Jakości, i jest definiowana jako: *zdolność produktu softwarowego do bycia przystępnym, operowalnym oraz atrakcyjnym, gdy jest używany w określonych warunkach*.

W niniejszym rozdziale przedstawiono normy i definicje użyteczności oraz dostępności, metody zapewnienia użyteczności, miary użyteczności systemów interakcyjnych oraz syntetyczna miara użyteczności systemów interakcyjnych.

### 2.4.1. Normy i definicje użyteczności i dostępności

Powszechnie uznawaną organizacją normalizacyjną jest we współczesnym świecie International Standardisation Organization (ISO). Również inne organizacje, takie jak International Electrotechnical Commission (IEC) i World Wide Web Consortium (W3C), która powołało bardzo ściśle współpracujące z ISO Web Accessibility Initiative (WAI) opracowują zasady powszechnie uznawane przez władze wielu krajów, a nawet przez ponadnarodowe organizacje, takie jak Wspólnota Europejska [Wagłowski 2004]. Normy te wyznaczają określone standardy, których przestrzeganie jest coraz powszechniej wymagane w wielu rozwiniętych krajach dla systemów udostępnianych przez instytucje i organizacje publiczne i stosowanych w takich sferach, jak edukacja czy służba zdrowia. W Polsce mimo że normy ISO zostały bezpośrednio przyjęte przez Polski Komitet Normalizacyjny (często nawet bez ich tłumaczenia na język polski) wymaganie ich stosowania nie uzyskało do tej pory bezpośredniego zapisu w Polskim prawie [Wagłowski 2004], a ich praktyczne przestrzeganie jest do tej pory jedynie sporadycznie weryfikowane.

Organizacja ISO przyjęła co najmniej kilka norm dotyczących jakości wykorzystania produktu, interfejsu i interakcji, dokumentacji, procesu wytwarzania oraz ergonomii [URG 2002]. Należą do nich przede wszystkim normy:

- ISO/IEC 9126: *Software engineering – Product quality* (Inżynieria oprogramowania – Jakość Produktu),
- PN-EN ISO 9241: *Wymagania ergonomiczne dotyczące pracy biurowej z zastosowaniem terminali wyposażonych w monitory ekranowe* (VDT) (ang. Ergonomic requirements for office work with visual display terminals (VDTs)),
- ISO 13407: *Human-centred design processes for interactive systems* (Zorientowany na człowieka proces projektowania systemów interakcyjnych),
- ISO 13406: *Ergonomic requirements for work with visual displays based on flat panels* (Wymagania ergonomiczne pracy z płaskimi wyświetlaczami panelowymi),
- ISO AWI 18789: *Ergonomic requirements and measurement techniques for electronic visual displays* (Wymagania ergonomiczne i techniki pomiaru dla elektronicznych wyświetlaczy),
- ISO/IEC 14754: *Pen-based interfaces – Common gestures for text editing with pen-based systems* (Interfejsy rysikowe – Powszechnie używane gesty w edycji tekstów z wykorzystaniem systemów bazujących na rysikach),
- ISO 11064: *Ergonomic design of control centres* (Ergonomiczne projektowanie centrów sterowania).

Wymienione normy składają się z wielu części, np. norma EN ISO 9241 z siedemnastu, ISO 11064 z ośmiu, ISO AWI 18789 z siedmiu, przy czym pewne części nowszych norm aktualizują lub zastępują wcześniejsze normy lub ich części, np. ISO AWI 18789 aktualizuje lub znosi PN-EN ISO 9241 Części 3, 7 i 8, oraz ISO 13406.

Normy ISO nie są oczywiście jedynymi obowiązującymi w różnych krajach normami i przepisami regulującymi problemy tworzenia i wykorzystania interfejsów użytkownika. Należy tutaj wymienić również inne regulacje, będące często pewnymi wstępnymi etapami poprzedzającymi przyjęcie ich jako normy międzynarodowe, np.:

- ANSI/INCITS-354: *Common Industry Format for Reporting Usability Results* (Wspólny Przemysłowy Format Raportowania Rezultatów Użyteczności),
- IEC CDV (Committee Draft for Vote) TR (Technical Report) 61997: *Guidelines for the user interfaces in multimedia equipment for general purpose use* (2000) (Wytyczne dla interfejsów użytkownika w urządzeniach multimedialnych powszechnego użytku),
- WAI WCAG 2.0: *Web Accessibility Initiative of the W3C – Web Content Accessibility Guidelines* (zalecenia dostępności dla webowej zawartości).

Definicje użyteczności prezentowane w ramach norm ISO składają się najczęściej z kilku różnych elementów składowych; należą do nich m.in.:

- skuteczność (ang. *effectiveness*) – dokładność i kompletność osiągnięcia określonych celów (PN-EN ISO 9241-11),
- wydajność (ang. *efficiency*) – wydatkowane zasoby w odniesieniu do dokładności i kompletności osiągnięcia określonych celów przez użytkowników (PN-EN ISO 9241-11),
- satysfakcja (ang. *satisfaction*) – wolność od dyskomfortu i pozytywne nastawienie w odniesieniu do użycia produktu (PN-EN ISO 9241-11),

- kontekst użycia (ang. *context of use*) – użytkownicy, zadania, wyposażenie (hardware, software i materiały) oraz fizyczne i socjalne środowiska, w których produkt jest używany (PN-EN ISO 9241-11),
- system wykonawczy (ang. *work system*) – system składający się z użytkowników, środowiska, zadań i fizycznego i socjalnego środowiska, przeznaczony do realizacji określonych celów (PN-EN ISO 9241-11),
- użytkownik (ang. *user*) – osoba, która wchodzi w interakcję z produktem (PN-EN ISO 9241-11),
- cel (ang. *goal*) – założony rezultat (PN-EN ISO 9241-11),
- zadanie (ang. *task*) – działanie wymagane do osiągnięcia zamierzonego celu (PN-EN ISO 9241-11),
- produkt (ang. *product*) – część wyposażenia (hardware, software i materiały), dla którego specyfikuje się bądź wyznacza użyteczność (PN-EN ISO 9241-11),
- przystępność (ang. *understandability*) – zdolność produktu softwerowego do umożliwienia użytkownikowi dokonania oceny przydatności oraz zrozumienia w jaki sposób i w jakich warunkach produkt może zostać użyty (ISO/IEC 9126-1),
- przyswajalność (ang. *learnability*) – zdolność produktu softwerowego do umożliwienia użytkownikowi nauczania się jego obsługi (ISO/IEC 9126-1),
- operowalność (ang. *operability*) – zdolność produktu softwerowego do umożliwienia użytkownikowi opanowania sposobu jego obsługi (ISO/IEC 9126-1),
- atrakcyjność (ang. *attractiveness*) – zdolność produktu do stania się atrakcyjnym dla użytkownika, odnosi się ona do tych atrybutów softwaru, które mają go uatrakcyjnić z punktu widzenia użytkownika, np. kolorystyka czy projekt graficzny (ISO/IEC 9126-1).

Normy PN-EN ISO 9241-11 i ISO/IEC 9126-1 koncentrują się na specyfikacji wymagań użyteczności konkretnego produktu, natomiast norma ISO 13407 skupia swoją uwagę na procesie wytwarzania produktów softwarowych i stanowi bazę do metodologii projektowania zorientowanego na użytkownika (ang. *User Centered Design* – UCD). Proces ten składa się z pięciu faz, z których pierwszą jest faza *planowania procesu zorientowanego na użytkownika*. Cztery pozostałe tworzą pętlę: *specyfikacja kontekstu użycia*, *specyfikacja wymagań organizacyjnych i użytkownika*, *wytworzenie produktu*, *ewaluacja projektu w odniesieniu do wymagań*. Do każdej z tych faz można przyporządkować specyficzne metody zapewnienia użyteczności. Na przykład w fazie *planowania procesu zorientowanego na użytkownika* wykorzystuje się następujące metody: planowanie użyteczności, spotkanie wstępne, analiza kontekstu użytkownika oraz analiza konkurencji. Z kolei w fazie *ewaluacji projektu w odniesieniu do wymagań* wyróżnia się następujące metody: ocena heurystyczna, inspekcje – listy kontrolne, testy z udziałem użytkowników i kwestionariusze ankietowe. Dwie spośród tych metod są ze względu na ich dużą przydatność najczęściej stosowanymi metodami badania użyteczności gotowych produktów lub ich prototypów, a mianowicie są to: ocena heurystyczna i testy z udziałem użytkowników, a wśród nich – badanie z wykorzystaniem urządzeń do śledzenia wzroku.

Oczywiście formalne unormowania są najczęściej efektem wcześniejszych prac badawczych, dlatego w literaturze można znaleźć wiele innych definicji użyteczności. Jednym z najbardziej znanych i zasłużonych specjalistów w dziedzinie użyteczności systemów interakcyjnych jest Jakob Nielsen, który definiuje użyteczność jako atrybut jakości oceniający łatwość użytkowania interfejsu przez użytkownika oraz również jako pojęcie odnoszące się do metod poprawiających łatwość jego użycia, składający się z pięciu składników: nauczalności, wydajność, zapamiętywanie, błędy i satysfakcja [Nielsen 1995]. Trzy z tych elementów zostały już określone. Pozostałe dwa zostały określone w następujący sposób:

- zdolność zapamiętywania (ang. *memorability*) – dotyczy łatwości powtórnego wykorzystania interfejsu po pewnej przerwie w jego użytkowaniu [Nielsen 1995],
- błędy (ang. *errors*) – odnosi się do całego kompleksu problemów związanych z liczbą popełnianych przez użytkownika błędów, obsługą błędów przez system oraz łatwością, z jaką można się z nich wydobyć [Nielsen 1995].

Inny specjalista z dziedziny systemów interakcyjnych, Steve Krug, określa użyteczność jako przekonanie, że coś pracuje właściwie w przypadku używania przez tzw. osobę o przeciętnych (lub nawet poniżej przeciętnych) zdolnościach i doświadczeniu – bez względu na to czy chodzi o używanie strony sieci Web, czy też samolotu myśliwskiego albo drzwi obrotowych [Krug 2000].

## 2.4.2. Metody zapewnienia użyteczności

Prezentując metody zapewnienia użyteczności, można posłużyć się przynajmniej dwoma kryteriami ich podziału. Pierwsze, najbardziej popularne kryterium, dzieli metody zapewnienia użyteczności w odniesieniu do etapu realizacji projektu, w którym stosuje się określoną metodę [UsabilityNet 2006], przy czym wyróżnia się tu następujące etapy: planowanie i analiza możliwości wykonania, analiza wymagań, projektowanie, implementacja, testowanie i pomiary, oraz użytkowanie i doskonalenie „po wydaniu” (ang. *post release*). Drugie kryterium dzieli metody w odniesieniu do ich charakteru i wyróżnia następujące [Hom 1998] metody: inspekcyjne, wywiadowcze (ang. *inquiry methods*), prototypowania, testowania oraz pozostałe metody, który to podział można uzupełnić o metody modelowania kognitywnego [Newman 1994]. Oba te podziały przedstawiają odpowiednio tabela 1 i 2.

Porównując metody przedstawione w obu tabelach (tabela 1 i 2), można zauważyć, że niektóre metody z tabeli 1 są bardziej ogólne niż metody przedstawione w tabeli 2. Dla przykładu strukturalizacja informacji zawiera m.in. metodę sortowania kart, czyli jedną z metod wyszczególnionych jako pozostałe w tabeli 2.

Zostaną przedstawione wybrane metody zapewnienia użyteczności, które można zastosować w ocenie użyteczności spersonalizowanych interfejsów użytkownika: modelowanie na poziomie wciskania klawiszy, procesualny model człowieka, wywiad, grupa fokusowa, ocena heurystyczna, wzorce projektowania, ewaluacja diagnostyczna, ankieta, subiektywna ocena i zdalna ocena. Należą do nich przede wszystkim metody przyporządkowane do etapu realizacji projektu testowanie i pomiary, oraz etapu użyt-

Tabela 1. Podział metod zapewnienia użyteczności ze względu na etap realizacji projektu  
[UsabilityNet 2006]

| Planowanie i analiza możliwości wykonania | Analiza wymagań             | Projektowanie                   | Implementacja          | Testowanie i pomiary                                    | Użytkowanie i doskonalenie „po wydaniu” |
|-------------------------------------------|-----------------------------|---------------------------------|------------------------|---------------------------------------------------------|-----------------------------------------|
| Rozpoczęcie                               | Zbieranie informacji        | Przewodniki projektowania       | Przewodniki stylów     | Ewaluacja partycypacyjna                                | Testowanie po wydaniu                   |
| Spotkanie udziałowców                     | Strukturalizacja informacji | Prototypowanie na papierze      | Szybkie prototypowanie | Ewaluacja diagnostyczna                                 | Ocena subiektywna                       |
| Analiza kontekstu                         | Objaśnienia zadań           | Ocena heurystyczna              |                        | Ewaluacja heurystyczna                                  | Wywiady z użytkownikami                 |
| ISO 13407                                 | Akceptacja wymagań          | Równoległe projektowanie        |                        | Ewaluacja subiektywna                                   | Zdalna ocena                            |
| Analiza konkurencji                       |                             | Tworzenie scenariusza           |                        | Analiza krytycznych wyjątków                            |                                         |
|                                           |                             | Ocena prototypu                 |                        | Ankiety oceniające postawy wobec projektowanego systemu |                                         |
|                                           |                             | Czarnoksiężnik z krainy Oz      |                        |                                                         |                                         |
|                                           |                             | Wzorce projektowania interfejsu |                        |                                                         |                                         |



Tabela 2. Podział metod zapewnienia użyteczności ze względu na ich charakter

| Metody modelowania kognitywnego            | Metody inspekcyjne                  | Metody wywiadowcze               | Metody testowania                                       | Pozostałe              |
|--------------------------------------------|-------------------------------------|----------------------------------|---------------------------------------------------------|------------------------|
| Modelowanie na poziomie wciskania klawiszy | Lista kontrolna zasad projektowania | Sesje protokolowane              | Ewaluacja partycypacyjna                                | Diagramy związków      |
| Procesualny model człowieka                | Analiza spójności                   | Grupy fokusowe                   | Ewaluacja diagnostyczna (w tym ze śledzeniem wzroku)    | Benchmarking           |
|                                            | Ewaluacja heurystyczna              | Wywiady kontekstowe              | Ewaluacja subiektywna                                   | Persona                |
|                                            | Inspekcja użyteczności              | Wywiady grupowe i grupy fokusowe | Analiza krytycznych wyjątków                            | Szybkie prototypowanie |
|                                            | Inspekcja pluralistyczna            | Studia etnograficzne             | Ankiety oceniające postawy wobec projektowanego systemu | Sortowanie kart        |
|                                            | Analiza kontekstu użycia            | Kwestionariusze                  |                                                         | Ocena nauczalności     |
|                                            | Analiza funkcjonalna                | Logi z autokomentarzem           |                                                         | Tajne głosowanie       |
|                                            | Inspekcja dotrzymania standardów    | Zrzuty ekranowe                  |                                                         |                        |
|                                            | Inspekcja spójności                 |                                  |                                                         |                        |
|                                            | Inspekcja kognitywny                |                                  |                                                         |                        |

kowanie i doskonalenie. Podstawą dokonania niniejszego wyboru jest przydatność tych metod w określaniu użyteczności rekomendowanych interfejsów użytkownika.

Jednym z najprostszych modeli kognitywnych jest **modelowanie na poziomie wciśnięcia klawiszy** (ang. *keystroke-level modelling*) [Newman 1996]. Pozwala ono przewidzieć szybkość pracy użytkownika z interfejsem z zastosowaniem klawiatury do wprowadzania danych i komend. Według tego modelu całkowity czas realizacji komendy jest sumą czasu jej wprowadzenia (średnio 2 znaki/s) i czasu reakcji systemu. Model na poziomie przyciskania klawiszy opracowany przez Newella i innych mówi, że w pewnych miejscach (np. po każdym wyrazie) użytkownik robi przerwy na tzw. przygotowanie umysłowe (średnio 1,35 znaki/s).

**Procesualny model człowieka** [Newman 1996] wyróżnia następujące podsystemy przetwarzania informacji: akustyczny, wizualny, artykulacyjny i ruchowy. Model ten zakłada wieloprocessorowość i równoległość wykonywania działań każdego z tych podsystemów. Podsystemy te cechuje również zróżnicowana szybkość pracy oraz osobne pamięci: robocza i trwała. W niniejszym modelu zakłada się, że średnio pojedynczy cykl wykonania nieskomplikowanego działania obejmuje następujące fazy: percepcyjną (akustyczną i wizyjną) trwającą 100 ms, kognitywną trwającą 70 ms i motoryczną trwającą również 70 ms. Badania przeprowadzone nad graczami gier komputerowych wykazały możliwości wytrenowania pozwalające na znaczne skrócenie czasu trwania tych cykli. Wyniki tych badań z powodzeniem mogą zostać rozszerzone na wszelkich użytkowników, którzy intensywnie trenują wykonywanie pewnych powtarzalnych operacji. Dotyczy to z pewnością doświadczonych kierowców, u których czas realizacji pewnych czynności jest o wiele krótszy, niż suma średniego czasu z trzech faz określających cykl tej czynności. Podobnie rzecz się ma w odniesieniu do obsługi wszelkich interfejsów systemów, które zostały opanowane przez użytkownika w bardzo wysokim stopniu.

**Wywiad** polega na zadawaniu częściowo zestrukturalizowanych pytań osobie badanej. Badaniom mogą podlegać zarówno użytkownicy, jak również inne osoby zaangażowane w pracę systemu, takie jak: jego dysponenci, eksperci dziedzinowi oraz osoby współpracujące z użytkownikami systemu. Zaletami tej metody są niskie koszty w zbieraniu danych (choć w literaturze można znaleźć również przeciwne zdanie [Pearrow 2002]) oraz znaczna efektywność w identyfikacji potrzeb użytkowników. Wadami są: przemilczenie wszelkich rzeczy, których użytkownicy nie zamierzają ujawnić bądź nie są ich świadomi oraz trudności z zapisem informacji podczas wywiadu (trudności te mogą być pokonane poprzez nagranie audio lub wideo [UsabilityNet 2006]).

**Grupa fokusowa** jest metodą, w której od 6 do 10 użytkowników bierze udział w moderowanej sesji, wymieniając swoje uwagi na temat określonego systemu [Pearrow 2002]. Sesje powinny trwać od jednej do trzech godzin i powinny być prowadzone z różnymi grupami, aż do otrzymania powtarzalnych wyników. Zaletami tej metody są: możliwość szybkiego odkrycia różnych błędów i niedogodności w pracy systemu oraz dostarczenie wielu istotnych danych w relatywnie krótkim czasie. Wadami tej metody są: konieczność zatrudnienia doświadczonego moderatora, niebezpieczeństwo skupienia się grupy na temacie pobocznym wywołanym przez jednego z uczestników dysku-

sji oraz niespójności pomiędzy tym, co użytkownicy są skłonni powiedzieć w szerszym gronie, a ich rzeczywistymi doświadczeniami w pracy z systemem [UsabilityNet 2006].

Bardzo popularną metodą, którą z powodzeniem można zastosować również na etapie testowania i pomiarów, jest **ocena heurystyczna**. Metoda ta została opracowana przez Jakoba Nielsena [Pearrow 2002], który określił dziesięć bardzo popularnych heurystyk badania użyteczności systemów interakcyjnych. Ewaluacja dokonywana jest przez grupę kilku ekspertów, przy czym najlepszy stosunek nakładów do efektów osiągany jest wówczas, gdy ich liczba wynosi 5. W ramach badania eksperci zaznaczają ile razy oraz w jakich sytuacjach oceniany system narusza określoną zasadę. W ramach tego badania można również określać stopień naruszenia określonej zasady, počawszy od nieznacznego (kosmetycznego) aż po krytyczny. Główną zaletą tej metody jest jej szybkość i taniość zastosowania, a ponadto to, iż osiągane rezultaty są uznawane za bardzo cenną informację zwrotną dla projektantów, którą można stosować zarówno na etapie projektu, jak i implementacji oraz testowania i pomiarów. Metoda ta umożliwia zidentyfikowanie problemów, których nie można wykryć z zastosowaniem metod badania wykonania (np. testów z użytkownikiem) [UsabilityNet 2006]. Opracowane przez Jakuba Nielsena heurystyki brzmią następująco [Pearrow 2002]:

1. Widoczność stanu systemu.
2. Dostosowanie systemu do świata rzeczywistego.
3. Kontrola versus swoboda użytkownika.
4. Spójność i standardy.
5. Ostrzeganie przed błędami.
6. Rozpoznawanie a nie przypominanie.
7. Elastyczność i wydajność.
8. Estetycznie i oszczędnie.
9. Wsparcie użytkowników w rozpoznawaniu, diagnozowaniu i radzeniu sobie z błędami.
10. Pomoc i dokumentacja.

Ponieważ heurystyki Nielsena powstały jeszcze przed rozpowszechnieniem aplikacji internetowych, w późniejszym okresie opracowano dodatkowe zasady heurystyczne odnoszące się specjalnie do tego typu aplikacji [Pearrow 2002]. Metoda oceny heurystycznej umożliwia w miarę swobodne opracowywanie heurystyk, ograniczając się jedynie do zapewnienia, aby zrozumienie ich było powszechne dla ekspertów oceniających system oraz projektantów systemu.

Kolejną, szczególnie istotną dla problematyki poruszanej w niniejszej pracy metodę zapewnienia użyteczności stosowaną na etapie projektowania stanowią tzw. **wzorce projektowania interfejsu** (ang. *interface design patterns*) [Tidwell 2005]. Metoda ta oferuje rozwiązania często pojawiających się problemów projektowania interfejsu użytkownika. Metoda ta jest relatywnie nowa, nie została więc jeszcze dokładnie zweryfikowana przez wielką liczbę praktyków projektowania [UsabilityNet 2006], uważa się jednak, że pozwala ona na przejście w miarę łatwy sposób od ogólnych wymagań użytkownika do softwarowej implementacji interfejsu.

Wzorce projektowania interfejsu mogą zostać zastosowane na etapie analizy możliwości wykonania – w określaniu wymagań oraz na etapie projektowania – w tworzeniu prototypów. Wzorce mogą być przydatne wielorako: mogą dostarczać idei projektowania, mogą dostarczać dobrych przykładów rozwiązania określonych problemów tworzenia interfejsu, określać sposób komunikacji (mówi się tu nawet o języku wzorców projektowania) w dyskusjach ze wszystkimi osobami zaangażowanymi w realizację projektu, mogą również dostarczać objaśnień, dlaczego określony wzorzec ma odpowiednią użyteczność.

**Przewodniki stylów** [UsabilityNet 2006] są stosowane do zapewnienia spójności wyglądu i obsługi interfejsu użytkownika. Ich wykorzystanie można zaliczyć do stosowania tzw. dobrych praktyk w tworzeniu interfejsu użytkownika, które przyczyniają się do zapewnienia spójności pomiędzy poszczególnymi ekranami (oknami aplikacji), ograniczenia liczby możliwych do popełnienia błędów, redukcji czasu implementacji interfejsu, co sumarycznie poprawia jakość interfejsu. Przewodniki stylów obejmują opracowane przez różnych specjalistów lub firmy komputerowe przewodniki dla GUI, np. firmę Microsoft [Microsoft 1999] lub nowsze przewodniki stylów do aplikacji pod systemem MS Windows Vista [Microsoft 2007], a także dla interfejsów webowych [Nielsen 2003].

**Ewaluacja diagnostyczna**, zwana również testami wykonania z udziałem użytkowników, jest najbardziej rozbudowaną, najkosztowniejszą, ale i przy tym najbardziej efektywną metodą wykorzystywaną w UCD [Pearrow 2002]. Badanie polega w skrócie na obserwacji użytkowników realizujących najczęściej kilka zadań, które określone zostają na podstawie analizy najbardziej charakterystycznych zadań realizowanych za pomocą systemu z wykorzystaniem określonego interfejsu użytkownika [Hackos 1998]. Badania przeprowadzane są przez zespół badawczy w celu identyfikacji problemów użyteczności. Badania poprzedzane są właściwym doбором badanych użytkowników, np. przez wypełnienie odpowiednich kwestionariuszy, przebieg badania rejestrowany jest na wideo, bardzo często użytkownicy proszeni są o myślenie na głos podczas realizacji zadań, a na zakończenie organizowana jest odprawa (ang. *debriefing*), podczas której badani użytkownicy mogą zadawać dowolne pytania dotyczące badania oraz zadawane są im nieformalne pytania dotyczące systemu. Metoda ta ma, oprócz wymienionych, następujące zalety: nawet niewielka liczba badanych użytkowników (zwykle 5–12) umożliwia zidentyfikowanie większości problemów użyteczności, zidentyfikowane zaś problemy są problemami rzeczywistymi, w odróżnieniu od zidentyfikowanych innymi metodami. Główną wadą jest problem z doбором odpowiednich użytkowników.

Badania prowadzone podaną metodą w ostatnich kilku latach w Laboratorium Jakości Oprogramowania (LJO) na Wydziale Informatyki i Zarządzania Politechniki Wrocławskiej wykazały przydatność metody w identyfikacji problemów użyteczności pojawiających się podczas realizacji przygotowanych zadań [Sobecki 2007b]. Problemy te wynikały najczęściej z niezrozumienia terminologii wykorzystywanej przez system, a także z powodu umieszczania elementów interakcyjnych w niestandardowych miejscach i w formie niezrozumiałej dla użytkownika. Problemy okazywały się większe, gdy

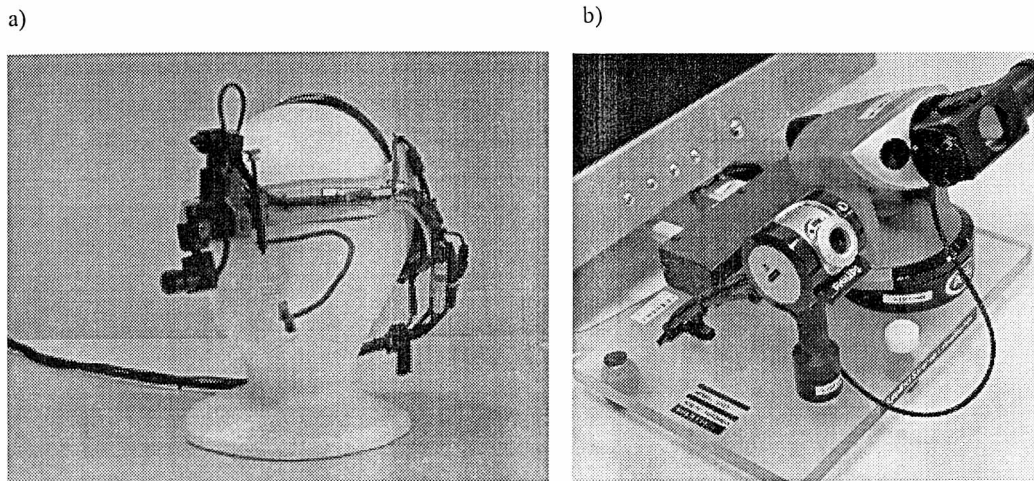
w formułowaniu zadania użyto języka bardziej potocznego od zastosowanego w systemie.

Badania, prowadzone przede wszystkim nad systemami webowymi w obszarze realizacji zadań polegających głównie na wyszukiwaniu informacji w określonym serwisie, pokazały spore różnice w metodzie obsługi przyjętej przez użytkowników o pewnym doświadczeniu w pracy z takimi systemami (np. użytkowników o wykształceniu informatycznym) i pozostałych użytkowników. Ci pierwsi preferowali wyszukiwanie na podstawie wyszukiwarek kontekstowych lub, w przypadku niepowodzenia, korzystanie z globalnej wyszukiwarki np. Google. Drudzy natomiast preferowali nawigację po odśladzaczach hipertekstowych.

Jedną z technik testowania wykonania (inaczej: mierzenia wykonania – ang. *performance measurement*) jest śledzenie ruchów gałki ocznej (ang. *eye-tracking*) [Usability-Net 2006] lub inaczej śledzeniem wzroku (ang. *gazetracking*). Technika ta pozwala na w miarę precyzyjne (w zależności od zastosowanej technologii) wyznaczenie gdzie, kiedy i jak długo patrzy użytkownik w odniesieniu do miejsca na ekranie monitora. Wśród tych technologii wyróżnia się elektrody naskórne (np. EOG), oznaczone szklą kontaktowe, kamery przetwarzające obraz oraz urządzenia śledzące odbicia (np. rogówkowe). Elektrookulografia (ang. *electro-oculography* – EOG) bazuje na gradiencie aktywności metabolicznej pomiędzy przodem i tyłem gałki ocznej [Aslin 2004] mierzonej przez elektrody otaczające jedno lub oboje oczu. Ruchy oka powodują, że na elektrodach mierzona jest różnica potencjału elektrycznego, która jest liniowo zależna od kąta patrzenia (w zakresie  $\pm 30^\circ$ ). Natomiast metoda refleksu rogówkowego (ang. *corneal-reflection* – CR) bazuje na bardzo prostej zasadzie, według której rogówka oka odbija światło pochodzące z punktu fiksacji (ang. *fixation point*). Dla kamery ulokowanej zaraz za źródłem światła i nakierowanej na tęczówkę oka refleks pojawi się w środku źrenicy. W miarę przesuwania fiksacji w prawo, lewo, w górę lub w dół refleks zmieni odpowiednio swoje położenie w stosunku do środka źrenicy [Aslin 2004]. W praktyce stosuje się zestaw kamery z lustrem półprzepuszczalnym, jak w przypadku urządzenia Head Mounted Tracker (HMT) firmy ASL [ASL 2005] (rys. 3), które montowane jest na głowie użytkownika. Spotyka się także zestawy wykorzystujące kamerę z dodatkowym oświetleniem światłem bliskopodczerwonym, tak jak w przypadku urządzenia Pan/Tilt (P/T) [ASL 2006], również firmy ASL (rys. 3) czy też podobnego urządzenia: Tobi ET-170.

Oprócz urządzenia do śledzenia wzroku, np. HMT lub P/T, wykorzystuje się także: odpowiedni sterownik tego urządzenia, np. ASL Eye-Track 6000 control unit, oprogramowanie do analizy skupienia wzroku GTAnaly oraz urządzenie do śledzenia ruchów głowy Head Tracker (HT) z urządzeniem sterującym Flock of Birds. Ponieważ wyposażenie laboratorium w sprzęt i oprogramowanie do śledzenia wzroku jest ciągle dość drogie (cena zestawu sięga obecnie 300–400 tysięcy złotych), a samo ich wykorzystanie wymaga sporego nakładu czasu na kalibrację z każdym użytkownikiem, ich zastosowanie ogranicza się wyłącznie do sytuacji, w których informacja o tym, gdzie w danej chwili patrzy użytkownik jest rzeczywiście niezbędna.

Należy jednak zauważyć, co wykazały badania przeprowadzone w LJO [Sobecki 2007b], iż w wielu przypadkach dopiero badanie z tego typu urządzeniem pozwala na



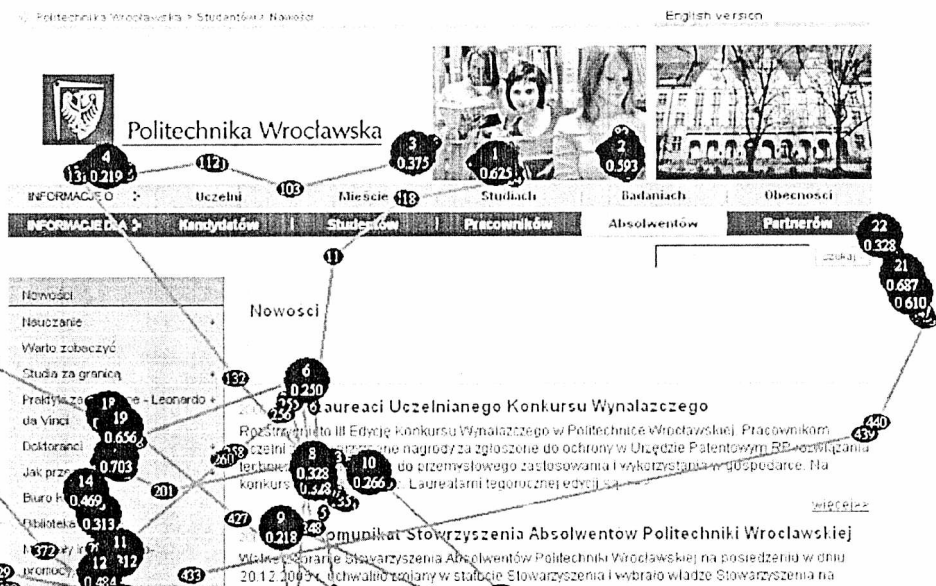
Rys. 3. Zdjęcia wykorzystywanych w LJO urządzenia HMT (a) i Pan/Tilt (b) – fot. J. Sobecki

wykrycie istniejących problemów z interfejsem. Bardzo często okazuje się, że występujące na ekranie informacje o stanie systemu lub nawet pewne elementy menu są przez wszystkich badanych użytkowników absolutnie niezauważane. Przykładowo badanie przeprowadzone w roku 2006 nad użytecznością nowo wprowadzanej wówczas wersji portalu Politechniki Wrocławskiej pokazało, że stan systemu, tzn. informacja, na jakiej stronie znajduje się obecnie użytkownik, jest dla użytkownika zupełnie niewidoczna. Wśród badanych pięciu użytkowników realizujących po 5 zadań, prawie żaden z nich nie skupił swojego wzroku na niniejszej informacji, co ilustruje rysunek 4. Na rysunku tym zaznaczono elipsą obszar, w którym znajduje się informacja o stanie systemu (tzn. bieżącej sekcji) linią zaznaczono ślad śledzenia wzroku składający się z punktów spojrzenia (zaznaczonych małymi ponumerowanymi czarnymi elipsami), jak i punkty dłuższej trwającej fiksacji (zaznaczone czarnymi kołami); jak widać, linie śledzenia wzroku zupełnie pomijają zaznaczony elipsą obszar z informacją o stanie systemu. Wniosek, jaki można wyprowadzić z tego badania jest taki, że stan systemu jest zupełnie niewidoczny dla użytkownika, gdyż leży poza obszarem (nawet powyżej logo), w którym użytkownik spodziewa się znaleźć jakąkolwiek informację dotyczącą tego systemu. Element ten powinien zostać przeniesiony w inne miejsce, np. poniżej menu poziomego.

Coraz częściej badania z wykorzystaniem sprzętu do śledzenia wzroku (szczególnie takie, które pozwalają użytkownikom na swobodne poruszanie głową, np. przez zastosowanie urządzenia HT) są łączone z innymi metodami badania użyteczności, tworząc metody hybrydowe, takie jak metoda UCEE [Voelkel 2006] łącząca ilościowe śledzenie wzroku z jakościowym wywiadem.

**Ankieta** polega na zadaniu użytkownikom standardowego zestawu pytań, przedstawionych na papierze, przez telefon lub w formie formularza elektronicznego. Podstawową zaletą tej metody jest szybkość i małe koszty przeprowadzenia ankiety wśród wielu





Rys. 4. Przykładowy ślad śledzenia wzroku z zaznaczonymi punktami fiksacji i punktami spojrzenia dla strony portalu Politechniki Wroclawskiej [LJO 2006]

użytkowników. Metoda ma jednak wiele wad stanowiących podstawę częstej krytyki, należą do nich: trudności opracowania odpowiedniego zestawu pytań, samodzielne odpowiadanie na pytania, słabo oddaje zachowania użytkowników, trudności w doborze reprezentatywnej próby użytkowników (gdyż to oni sami decydują o tym, czy będą uczestniczyć w ankiecie) oraz najcenniejsze pytania otwarte najczęściej pozostają bez odpowiedzi [UsabilityNet 2006].

**Subiektywna ocena** dostarcza informacji o tym, jak użytkownicy odbierają system, czyli o satysfakcji z użytkowania tak rozumianej, jak została ona zdefiniowana w normie 9241, część 11. Ocena ta realizowana jest z wykorzystaniem standaryzowanego kwestionariusza opinii. Zaletą tej metody jest dostarczenie bardzo istotnego indykatora zadowolenia z korzystania systemu. Jeżeli poziom satysfakcji z użytkownika systemu jest bardzo niski, może to stać się przyczyną problemów z absencją, szybką wymianą kadr oraz kolejnymi niezwiązanymi skargami z miejsca pracy. Badania satysfakcji stanowią uzupełnienie badań metodą ewaluacji diagnostycznej. Efektem działania tej metody jest zwykle lista cech systemu, które się podobają i które się nie podobają użytkownikom.

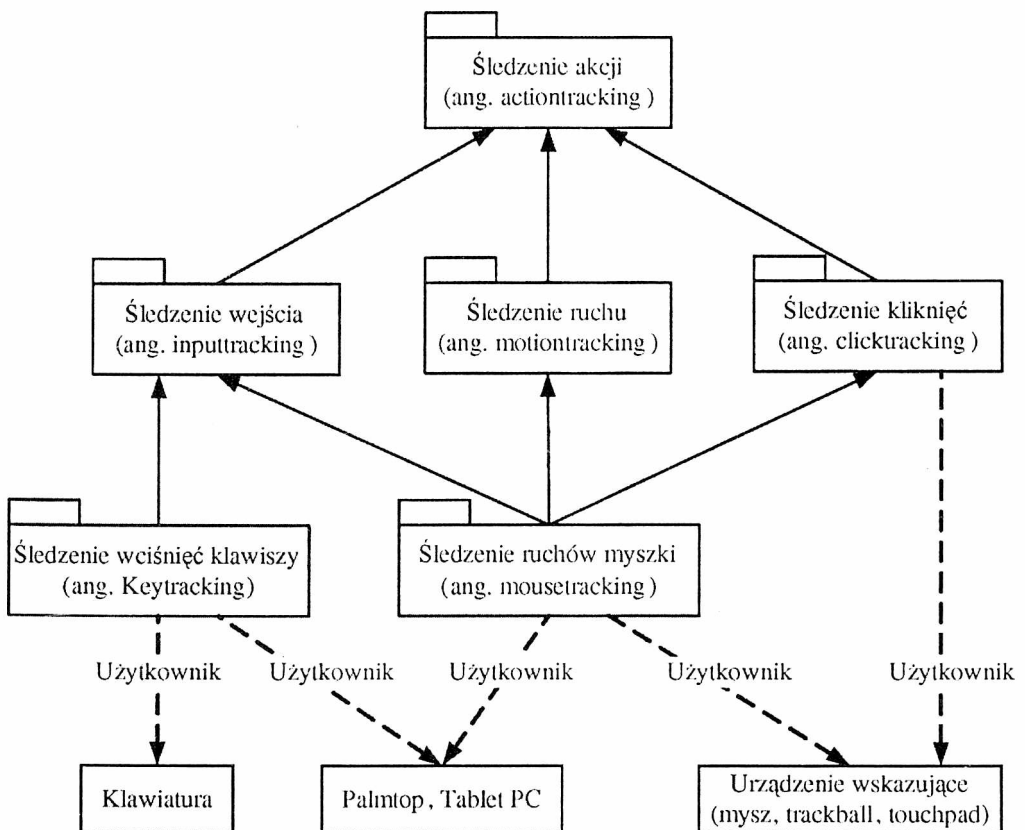
Alternatywą do badania kontrolowanego z udziałem użytkowników jest **zdalna ocena**, która może być realizowana jako analiza zapisywanych na serwerze plików logowań [Kobsa 2001] lub dokładniejszej metody actiontrackingu [Żatuchin 2008]. Pliki logowań zawierają zapisy różnorodnych akcji dokonywanych przez użytkowników w trakcie pracy z systemem. Tworzą one tzw. dane użytkownika [Kobsa 2001] służące m.in. do tworzenia wzorców zachowań użytkowników oraz obszarów ewentualnych problemów. Zaletą tej metody jest zapis historii użycia systemu, który pozwala na szybkie zbieranie danych bez specjalnego angażowania użytkowników. Wadą metody jest brak



informacji o powodach wykonywanych akcji oraz sposobach radzenia sobie użytkownika z ewentualnymi problemami [UsabilityNet 2006]. Druga metoda, actiontracking [Żatuchin 2008], w odróżnieniu od analizy plików logowań pozwala na automatyczne monitorowanie wszelkich podejmowanych przez użytkownika akcji na badanej stronie internetowej. Metoda ta, podobnie jak analiza plików logowań, należy do metod bez nadzoru. W ramach actiontrackingu można wyróżnić 3 typy akcji użytkownika (rys. 5):

- keytracking, czyli monitorowanie wprowadzanych na klawiaturze danych,
- moustracking, czyli monitorowanie ruchów myszy lub innego urządzenia wskazującego,
- clicktracking, czyli monitorowanie punktów w reprezentacji graficznej interfejsu użytkownika, w które nastąpiło kliknięcie.

System actiontrackingu ma wiele zalet w porównaniu z systemami do śledzenia ruchów gałki ocznej. Przede wszystkim jest tańszy w użyciu. Może być wykorzystywany w warunkach kontrolowanych, np. jako uzupełnienie tradycyjnych badań z udziałem użytkowników lub zupełnie bez nadzoru. Dzięki odpowiedniej funkcjonalności oprogramowania można badać nie tylko pojedynczych użytkowników, ale również w sposób wygodny połączyć np. mapy kliknięć dla wszystkich użytkowników w określonym



Rys. 5. Podział metod automatycznego monitorowania akcji użytkownika [Żatuchin 2008]

czasie. Wadą tych metod jest to, iż nie są one tak dokładne jak systemy gazetrackingu, a poziom zgodności ścieżki nie jest zbyt wysoki. Niemniej jednak tego typu narzędzie można zastosować do automatycznego badania użyteczności interfejsów adaptacyjnych.

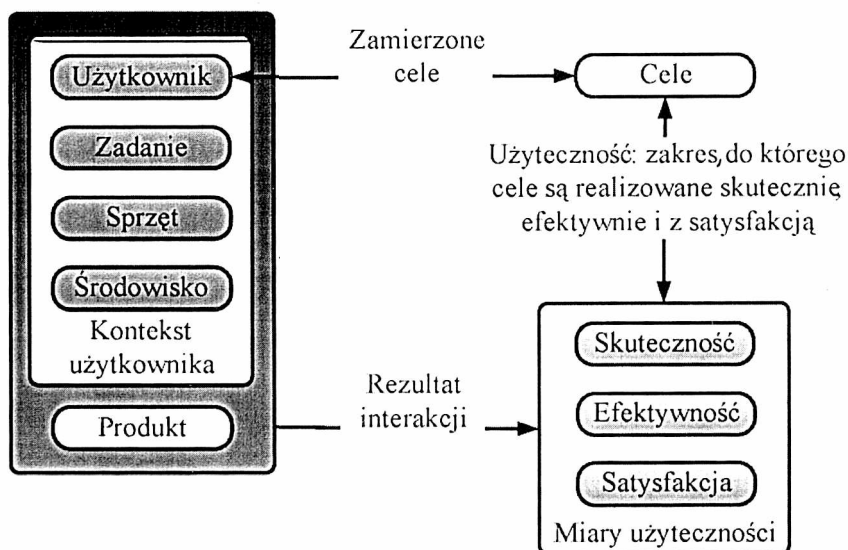
### 2.4.3. Miary użyteczności systemów interakcyjnych

W celu określenia miar użyteczności należy przeanalizować strukturę komponentów wchodzących w skład całego kontekstu użycia produktu (rys. 6). Na podstawie normy ISO 9241-11 przyjmuje się, że należy opracować przynajmniej po jednej mierze odpowiednio dla: **skuteczności, efektywności i satysfakcji**.

W tabeli 3 przedstawiono miary efektywności, skuteczności i satysfakcji dla wybranych celów użyteczności (ISO 9241-11).

**Miara skuteczności** powinna bezpośrednio odnosić się do dokładności i kompletności celów (lub podcelów) realizowanych za pomocą badanego produktu. Przykładowo, w sklepie internetowym byłoby oceniane zamówienie odpowiedniego towaru (dokładność) wraz z określeniem warunków zapłaty i jego dostarczenia (kompletność).

**Miara efektywności** odnosi się do wydatkowania zasobów, które mogą odnosić się zarówno do wysiłku fizycznego czy umysłowego użytkownika, jak i do zużytego czasu, pieniędzy czy innych środków materialnych. W przywołanym przykładzie sklepu internetowego można by zastosować miarę średniego czasu złożenia pojedynczego zamówienia, konieczność każdorazowego podawania określonych danych versus wykorzystanie zapamiętanych danych lub wreszcie sumaryczny transfer danych mający wpływ na czas obsługi a także cenę korzystania z systemu (np. przy korzystaniu z bezprzewodowej sieci GPRS).



Rys. 6. Struktura zapewnienia użyteczności (ISO 9241-11)

Tabela 3. Przykłady miary użyteczności (ISO 9241-11)

| Cel użyteczności                                               | Miary skuteczności                                                                                               | Miary efektywności                                                                                  | Miary satysfakcji                                      |
|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------|--------------------------------------------------------|
| Zaspokojenie potrzeb przez wyszkolonych użytkowników           | Liczba wykonanych trudnych zadań; procent wykorzystanych zaawansowanych funkcji                                  | Relatywna efektywność w porównaniu z ekspertem                                                      | Skala ocen satysfakcji z użycia zaawansowanych funkcji |
| Zaspokojenie potrzeb użytkowników bez szkolenia                | Procent akcji zakończonych powodzeniem za pierwszą próbą                                                         | Czas przeznaczony na pierwszą próbę <sup>1</sup> ; relatywna efektywność dla pierwszej próby        | Ocena użycia przez wolontariuszy                       |
| Zaspokojenie potrzeb dla rzadkiego lub średnio częstego użycia | Procent zadań wykonanych poprawnie po określonej przerwie w użyciu systemu                                       | Czas spędzony na ponowne uczenie się funkcji <sup>1</sup> ; liczba uporczywych błędów               | Częstotliwość ponownego użycia                         |
| Minimalizacja wsparcia                                         | Liczba odwołań do dokumentacji; liczba telefonów do pomocy on-line; liczba dostępów do pomocy systemu            | Produktywny czas <sup>1</sup> ; czas konieczny na przygotowanie do sprawdzianu <sup>1</sup>         | Skala ocen satysfakcji funkcji pomocy                  |
| Nauczalność                                                    | Liczba przyswojonych funkcji; Liczba użytkowników, którzy zdołali się przygotować do sprawdzianu                 | Czas konieczny na przygotowanie do sprawdzianu <sup>1</sup> ; relatywna efektywność podczas uczenia | Skala ocen dla łatwości nauczania                      |
| Tolerancja na błędy                                            | Procent błędów poprawionych lub wskazanych przez system; Tolerowana liczba błędów popełnionych przez użytkownika | Czas spędzony na poprawę błędów                                                                     | Skala ocen dla obsługi błędów                          |
| Czytelność                                                     | Liczba słów odczytanych poprawnie z normalnej odległości                                                         | Czas potrzebny na przeczytanie określonej liczby znaków                                             | Skala ocen dla wizualnego dyskomfortu                  |

<sup>1</sup> W tych przypadkach pomiaru należy dokonywać dla określonego poziomu efektywności.

**Miara satysfakcji** (PN-EN ISO 9241-11) – mierzy komfort użycia produktu i pozytywne nastawienie w odniesieniu do jego użytkowania. Metoda pomiaru może polegać na subiektywnej ocenie z wykorzystaniem pewnej skali w odniesieniu do następujących czynników: doświadczonego dyskomfortu, lubienia produktu, satysfakcji z użycia produktu, akceptacji nakładu pracy podczas wykonywania różnorodnych zadań i stopnia, do którego zostały spełnione cele użyteczności (takie jak efektywność czy nauczalność). Proponuje się również, aby miarą satysfakcji była liczba pozytywnych i negatywnych opinii. Dodatkowych informacji mogą dostarczyć pewne długookresowe miary dotyczące: absencji, obserwacji kognitywnego przeciążenia lub niedociążenia, raportów zdrowotnych, zmian stanowiska pracy przez użytkowników.

## 2.5. Personalizacja systemów webowych

Personalizacja jest dostosowaniem każdego produktu konsumenckiego do charakterystyki personalnej jego użytkownika. W przypadku systemów informacyjnych personalizacja definiowana jest jako cały zbiór działań dążących do dostarczenia treści i sposobu jej prezentacji dopasowany odpowiednio dla konkretnego użytkownika lub pewnej grupy użytkowników serwisu [RSI 2004]. Personalizacja różnorodnych serwisów webowych jest znana i wykorzystywana od wielu lat. Już w 2000 roku prawie wszystkie większe portale internetowe umożliwiały swoim użytkownikom możliwości personalizacji [Kowalczyk 2001], należy tutaj jednak dodać, że funkcjonalność ta nie cieszyła się wielkim zainteresowaniem polskich internautów i została od pewnego czasu mocno ograniczona, chociaż w modernizacji serwisu Onet.pl dokonanej w 2008 r. mechanizmy personalizacyjne zostały również uwzględnione.

W literaturze termin personalizacji często używany jest wymiennie z adaptacją lub adaptatywnością systemów webowych [Stephanidis 2001]. W niniejszej pracy głównym punktem zainteresowania jest dostosowanie (adaptacja) interfejsu użytkownika do jego indywidualnych potrzeb, jednak personalizacja może przysporzyć użytkownikowi wiele innych korzyści, wśród których należy wymienić m.in. [Adamczyk 2003], [Kobsa 2001]: możliwość dostosowania wyglądu strony i prezentowanych treści, umożliwienie prenumeraty określonych biuletynów, usprawnienie wyszukiwania oraz ułatwienie korzystania ze sklepów internetowych. Powodów, dla których personalizacja jest wykorzystywana przez dostawców serwisów internetowych może być bardzo dużo, a wśród nich można wymienić [Adamczyk 2001], [Sobecki 2004]: zapewnienie użyteczności i dostępności serwisów webowych, dotarcie z własnym przekazem do liczego audytorium, zdefiniowanie grupy docelowych odbiorców, zbudowanie własnej bazy adresowej, stworzenie serwisu będącego atrakcyjną platformą reklamową, zdobycie licznej grupy lojalnych klientów, uzyskanie przewagi nad firmami niestosującymi mechanizmów personalizacyjnych.

W systemach webowych można modyfikować następujące elementy: zawartość, strukturę i prezentację systemu [Kobsa 2001]. W implementacjach określonych systemów webowych można zastosować jedną, dwie lub wszystkie typy modyfikacji. W sy-

stemach służących przede wszystkim rozpowszechnianiu różnorodnych informacji, np. portali, najistotniejsza jest modyfikacja zawartości. W przypadku wielu systemów wspomagających nauczanie bardzo często wykorzystywana jest modyfikacja struktury, służąca najczęściej zmianie kolejności przeglądania materiału. Ostatni rodzaj modyfikacji służy zmianie sposobu prezentowanych informacji i stosowany może być z powodzeniem w każdym rodzaju systemów webowych.

W pracy [Kobsa 2001] wyszczególniono następujące typy adaptacji zawartości:

- dodatkowe (opcjonalne) wyjaśnienia, oferowane wyłącznie takim użytkownikom, którym mogą się one okazać przydatne, natomiast pozostają transparentne dla takich użytkowników, którzy ze względu na posiadaną wiedzę (lub jej brak) uznałyby je za zupełnie redundantne lub niezrozumiałe,
- dodatkowe informacje szczegółowe; funkcjonalne strony internetowe powinny być tworzone w taki sposób, aby najpierw prezentować informacje najbardziej ogólne, a dopiero dalej prezentować coraz bardziej szczegółowe informacje; taki styl prezentacji nazywany jest również odwróconą piramidą [Pearrow 2003],
- personalizowane rekomendacje; jest to najczęściej spotykane w agentach interfejsowych, które, jak agent wspomagający wyszukiwanie lokali mieszkalnych [Shearin 2001], pozwala na selekcję prezentowanej informacji pod kątem preferencji użytkownika, tego typu podejście można również znaleźć w portalach internetowych, które dają swoim użytkownikom możliwość wyboru sekcji, np. sport, giełda, pogoda także dla wybranego regionu (np. Lycos i Yahoo) [Sobecki 2001],
- podpowiedzi kontekstowe; reagują na określony kontekst interakcji i bazują z jednej strony na modelu użytkownika, a z drugiej na np. bieżącej pozycji kursora; pozwalają one np. na podświetlenie określonych sekcji lub wyświetlenie dodatkowych informacji w nowych oknach,
- język, waluta i jednostki miar; personalizacja tego typu, choć bardzo tradycyjna, może mieć decydujący wpływ na właściwe zrozumienie przez użytkownika przekazywanych treści.

Struktura serwisu webowego definiowana jest przez połączenia hipertekstowe lub szerzej hipermedialne, występujące w całym systemie. Ogólnie wyróżnia się dwa typy połączeń: kontekstowe (występujące wewnątrz prezentowanych treści) oraz bezkontekstowe (występujące jako niezależne od prezentowanych treści, np. przyciski wyboru opcji systemu webowego). Adaptacja struktury w pracy [Kobsa 2001] została podzielona na następujące typy:

- uboczna adaptacja struktury; ten typ modyfikacji występuje, gdy wraz z modyfikacją treści nie zostaje wyświetlony odsyłacz kontekstowy, który tam występuje,
- sortowanie odnośników; ten typ adaptacji występuje przede wszystkim w systemach rekomendacyjnych (agentach interfejsowych), np. w systemie *Letizia*, autonomicznym agencie wspomagającym przeglądanie sieci WWW [Liebermann 1997], który na podstawie modelu użytkownika sortuje wyszukane strony; tego typu adaptacja stosowana jest również w sklepach internetowych, dostarczając rekomendacji o książkach, płytach czy filmach na DVD (np. w internetowym megasklepie Amazon),

- zaznaczanie odnośników; dotyczy zarówno odnośników kontekstowych, jak i bezkontekstowych; najbardziej popularną formą zaznaczania odnośników jest rozróżnienie odnośników do stron już odwiedzonych i jeszcze nieodwiedzonych; w systemach adaptacyjnych wspomagających nauczanie na podstawie historii danego kursu różne kolory odnośników mogą wskazywać konkretnemu użytkownikowi działy, które już zostały przez niego przerobione, które należy powtórzyć lub pominąć itp.; w systemach informacji turystycznej na podstawie jego jawnych preferencji, można wskazywać takie strony, które będą dla użytkownika najciekawsze lub przeciwnie zdecydowanie nieinteresujące,
- ukrywanie i wyświetlanie odnośników; polega na zmianie postaci graficznej odnośnika do postaci normalnego tekstu lub odsyłacza hipertekstowego oraz usunięciu lub dodaniu łącza i hipertekstowego,
- włączanie i wyłączanie odnośników; polega wyłącznie na usunięciu lub dodaniu łącza i hipertekstowego,
- usuwanie i dodawanie odnośników; dotyczy odnośników bezkontekstowych i realizowane jest jako fizyczne dodanie lub usunięcie odsyłacza hipertekstowego.

Ostatnim elementem interfejsu, który może podlegać adaptacji, jest jego wygląd, co oznacza, że nie ulega zmianie jego zawartość ani struktura. Modyfikacja wyglądu może więc dotyczyć takich elementów systemu (graficznych, tekstowych czy dźwiękowych), które zmieniają wygląd interfejsu, lecz nie modyfikują jego zawartości treściowej. Dotyczy to również dźwięku. Modyfikacją prezentacji jest zatem podmiana ścieżki z podkładem muzycznym uprzyjemniającym interakcję z systemem, jednak nie byłoby nią usunięcie tekstu czytanego przez lektora. W portalach internetowych bardzo często spotykamy możliwość wyboru różnych wzorców kolorystycznych wyświetlanych treści; można również zmieniać krój i wielkość czcionek, a także rozmieszczenie, liczbę kolumn i/lub okien, w których są one prezentowane.

W miarę pełną gamę możliwości personalizacyjnych oferowały, przede wszystkim dla swoich zarejestrowanych użytkowników, serwisy internetowe Lycos i Yahoo. Są to portale z ponad dziesięcioletnią historią i z ugruntowaną pozycją na rynku globalnym. Występujące w nich możliwości personalizacyjne obejmowały głównie [Sobecki 2004] modyfikację zawartości informacyjnej i sposobu jej prezentacji, a zwłaszcza:

- wybór zawartości informacyjnej (patrz rys. 7),
- zmianę komunikatu powitania,
- edycję poszczególnych modułów,
- pasek szybkiego dodawania modułów w kolumnach,
- wybór schematu kolorów,
- szczegółowe dopasowywanie kolorów,
- wybór wyglądu strony (skórki) (patrz rys. 8),
- wybór pomiędzy schematem dwu- lub trzykolumnowym,
- częstotliwość odświeżania strony.



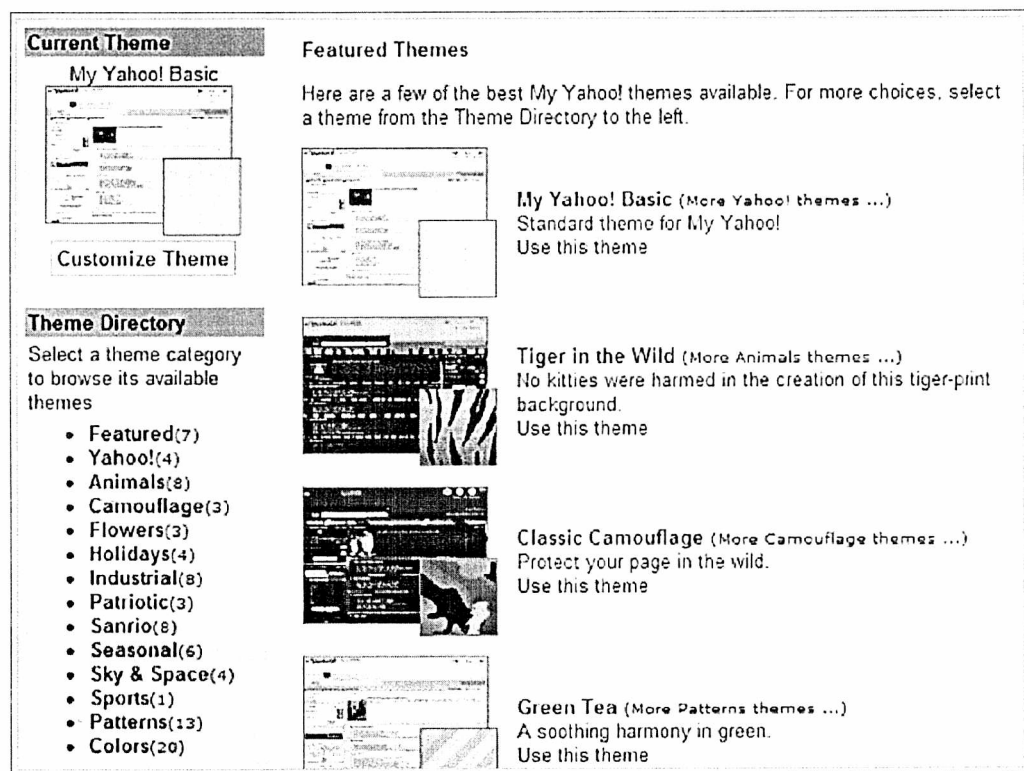


Rys. 7. Wybór zawartości informacyjnej w Yahoo! wraz z oknem podglądu

## 2.6. Perspektywy rozwoju systemów webowych

Pomimo iż system WWW, jaki znamy, liczy sobie dopiero kilkanaście lat, coraz częściej pojawiają się pomysły wprowadzania bardzo istotnych zmian do koncepcji webu, przy czym propozycje rozwoju oparte są na różnych przesłankach.

Jako propozycje o w miarę ugruntowanych pozycjach można wymienić Web 2.0 i Web 3.0. Pojęcie Web 2.0 zostało utworzone przez O'Reilly Media w 2003 roku i spopularyzowane w 2004 roku poprzez konferencję o tej nazwie [O'Reilly 2005]. Web 2.0 można scharakteryzować jako zastosowanie odpowiednich technologii umożliwiających całej społeczności Internetu łatwe udostępnianie, wymianę i współtworzenie informacji. W sferze technologicznej Web 2.0 wykorzystuje takie narzędzia, jak wiki, blogi, interfejsy XML do opublikowanych dokumentów; w sferze społecznej charakteryzuje się powszechnym tworzeniem informacji, użyciem folksonomii (ang. *folksonomy*), czyli swo-



Rys. 8. Wybór wyglądu strony (skórki) dla Yahoo!

bodnym kategoryzowaniem treści i wykorzystaniem słów kluczowych, tworzeniem społeczności wokół serwisów webowych, wykorzystywaniem kolektywnej inteligencji oraz wykorzystaniem otwartych licencji do prezentowanych treści oraz wykorzystywanego oprogramowania [O'Reilly 2005].

Jako pewien specyficzny typ systemu Web 2.0 można wymienić system Wiki, który uzyskał już dużą popularność oraz bardzo duże uznanie wśród prawie całej społeczności Internetu. Wiki reprezentują całą rodzinę różnego rodzaju systemów od Wikipedii, poprzez Wikitravel i Wikinews aż po wiele różnorodnych specjalizowanych systemów opartych na koncepcji i oprogramowaniu Wiki, np. Wiki-Wrocław. Wiki lub WikiWiki to nazwa systemu zarządzania zawartością (ang. *Content Management System* – CMS), czyli systemu udostępniającego możliwość wprowadzania, kasowania, poprawiania oraz wyszukiwania stron internetowych. Nazwa WikiWiki została zaczerpnięta z języka hawajskiego, gdzie zwrot *wiki wiki* oznacza *bardzo szybko* [Ward 2008].

Masowe korzystanie z blogów i systemów Wiki oraz kolaboratywne opracowywanie informacji powoduje, że Web 2.0 jest obecnie powszechnie stosowany. Nie oznacza to jednak, że ewolucja webu została zakończona. Od pewnego czasu pojawiają się dość krytyczne głosy dotyczące Web 2.0 zarzucające tej technologii petryfikację webu [Hard-

man 2008]. W 2006 r. krytyk koncepcji Web 2.0 Zeldman w swoim blogu użył pojęcia Web 3.0. Ponieważ jest to ciągle pojęcie w miarę nowe, jego znaczenie nie jest jeszcze precyzyjnie określone. Dla przykładu twórca WWW Berners-Lee powiedział o Web 3.0: *Myślę, że gdy masz nakładkę na skalowalną grafikę wektorową na Web 2.0, wszystko drga, łamie się i wygląda zamglone, natomiast w przypadku semantycznego Web-u (ang. Semantic Web) integrującego ogromne ilości danych, będziesz miał dostęp do niewyobrażalnego zasobu danych* [Shannon 2006]. Jak widać Web 3.0 utożsamiane jest przez wielu z webem semantycznym, który to termin został stworzony właśnie przez Bernersa-Lee w 2001 roku [Metz 2007]. Semantyczny web umożliwia automatyczne czytanie, analizę, przetwarzanie i przeszukiwanie zgromadzonych informacji w sposób analogiczny jak to robią ludzie. Obecnie semantyczny system web nie jest jedynie koncepcją, ale ma już wiele implementacji, np. *BlueOrganizer* firmy *AdaptiveBlue* będący wtyczką pełniącą rolę personalnego agenta [Wooldridge 1995], który w czasie przeglądania określonej strony „rozumie” jej treść i rekomenduje powiązane informacje pochodzące z innych serwisów webowych.

Obecnie oprócz wielu aplikacji są już zdefiniowane standardy opisu metadanych – *Recurse Description Framework (RDF)* i *Web Ontology Language (OWL)* [Metz 2007]. Formalna specyfikacja OWL wraz z syntaktyką podjęzyków OWL – OWL Lite i OWL DL, oraz semantyka RDF wraz z przyporządkowaniem syntaktyce abstrakcyjnej trójek RDF została przedstawiona w rekomendacji W3C [Patel-Schneider 2004]. Od pewnego czasu prowadzonych jest wiele prac nad reanotacją całych serwisów webowych polegającą na dodaniu różnego rodzaju metadanych czytelnych dla komputera (ang. *machine-readable*) do standardowych stron webowych czytelnych dla człowieka. Przytoczone standardy RDF i OWL, wykorzystywane są np. do reanotacji serwisu żywnościowego Yahoo! FOOD. Oczywiście takie przedsięwzięcie jest gigantyczne, a jego powodzenie w przyszłości trudno jest przewidzieć, gdyż z jednej strony możliwości formułowania pytań dotyczących informacji zawartej w takiej sieci wciąż bardzo wzrastają, jednak z drugiej strony wymagają one ogromnego nakładu pracy i czasu podczas tworzenia potrzebnych do tego niezmiernie wielkich liczby metadanych. Obecnie przewiduje się, np. w raporcie Gartnera [Harmelen 2008], że zastosowanie kombinacji semantycznego webu i Web 2.0 pozwoli na stopniowe i łagodne przejście od stosowania obecnego webu, poprzez nieskomplikowane semantycznie wykorzystanie prostych w użyciu aplikacji Web 2.0 do bardziej zaawansowanych, jednak również droższych technologii Web 3.0.

## 2.7. Podsumowanie

W niniejszym rozdziale przedstawiono podstawowe problemy systemów webowych, począwszy od opisu systemów interakcyjnych, hipertekstu i hipermediów, przez technologie systemów webowych oraz badanie ich użyteczności, po sposoby ich personalizacji i adaptacji oraz perspektywy rozwoju systemów webowych.

Przedstawiono najpierw definicje interakcji i systemów interakcyjnych, style interakcji oraz metody ich wyboru. Większość z systemów informacyjnych, będących przedmiotem licznych publikacji Autora, to systemy interakcyjne, wykorzystujące określone style interakcji, które są najczęściej aplikacją menu, klawiszy funkcyjnych i bezpośredniej manipulacji graficznej. Wkład Autora polega na rozszerzeniu typologii stylów interakcji o styl wirtualnej rzeczywistości oraz przedstawienie techniki multimedialnego stylu interakcji, który bazuje na wykorzystaniu niektórych parametrów rejestrowanych przez urządzenia do śledzenia wzroku, takich jak zmiany średnicy źrenicy. Zmiana średnicy źrenicy umożliwia identyfikowanie rozpoznania obrazu przez użytkownika [Sobecki 2008]. Problemy projektowania systemów interakcyjnych zorientowanych na użytkownika przedstawiono w pracach Autora [Sobecki 1999] i [Sobecki 2000]. Prace te zawierały propozycję określonego rozwiązania w przypadku projektowania systemów wspomagających nauczanie [Sobecki 1999] oraz tytułów multimedialnych [Sobecki 2000].

Hipertekst, multimedia, hipermedia i systemy webowe, są przedmiotem zainteresowania Autora od 20 lat. Dlatego większość publikacji autora dotyczy zastosowań tego typu systemów. Wśród najważniejszych można wymienić opis koncepcji i implementacji hipertekstowego systemu informacyjnego o problemach naukowych SWIP opublikowany jako rozdział w pracy zbiorowej [Choroś 1992].

Zainteresowania autora, dotyczące przetwarzania danych multimedialnych i technologii Internetowych, takich jak system WAIS, WWW i język HTML znalazły wyraz w publikacji m.in. w czterech rozdziałach pracy zbiorowej [Karpiak 1996]. Prace nad wykorzystaniem danych multimedialnych były kontynuowane, a ich wyniki przedstawione w publikacjach o wyszukiwaniu informacji multimedialnej w systemie WWW [Sobecki 1998] i o rozpoznawaniu osób na podstawie ich twarzy [Pieczyńska 2000].

Oryginalnym wkładem w rozwój systemów webowych było opracowanie przez Autora koncepcji i współautorstwo wdrożenia pierwszego w Polsce systemu udostępnienia katalogu książek w sieci WWW [Kołcio 1995]. Kontynuacją tych prac było uczestnictwo w realizacji projektu zintegrowanego katalogu książek dla kilku bibliotek uczelni technicznych w Polsce opartej na wykorzystaniu systemu WAIS, są to [Jakubowski 1996] i [Jakubowski 1997]. Doświadczenia z wykorzystania systemu wyszukiwania WAIS, zostały dalej rozwijane w koncepcji i implementacji systemu informacyjnego o raportach technicznych [Sobecki 1997] i [Sobecki 1997a]. Obecnie jedną z dominujących technologii w systemach webowych jest XML. Technologia ta została wykorzystana w implementacji do wymiany informacji w systemie rekomendacyjnym [Sobecki 2004b], ale również wykorzystana do modelowania użytkownika [Kazienko 2003] i interfejsu użytkownika [Sobecki 2003]. Architektura trójwarstwowa oraz różne technologie webowe, takie jak: cookies, PHP, Web 2.0 i inne, były wykorzystywane w ramach licznych implementacji webowych systemów rekomendacyjnych, takich jak [Sobecki 2004b], [Sobecki 2006a], [Rakowski 2006], [Sobecki 2007] i [Nguyen 2007].

Przedstawiono również definicje i normy użyteczności, metody zapewnienia użyteczności oraz ich typologie. Stanowią one niezbędne narzędzia zarówno do projektowania,

jak i realizacji oraz oceny systemów interaktywnych, ze szczególnym uwzględnieniem systemów webowych. Rekomendacja interfejsu użytkownika musi bazować na jego użyteczności, dlatego odpowiednie metody jej zapewnienia stanowią niezbędny element każdego systemu rekomendacji interfejsu użytkownika w adaptacyjnym systemie webowym. Rekomendacja interfejsu użytkownika jest realizowana zwykle automatycznie, tzn. bez udziału ekspertów, można jedynie wykorzystywać takie metody badania użyteczności, które na bieżąco ich nie angażują. Należą do nich głównie ankiety [Sobecki 2004b], w których użytkownicy sami oceniają satysfakcję z użytkowania określonego elementu. Jednak, zwłaszcza w przyszłości, należy rozważyć powszechniejsze zastosowanie metod zdalnej oceny, np. actiontrackingu [Żatuchin 2008]. Dzięki zastosowaniu metod zdalnej oceny, z odpowiednio opracowanym wzorcowym modelem postępowania użytkownika, wyznaczonym przez zastosowanie metod kognitywnych, można w sposób automatyczny określić użyteczność określonej instancji interfejsu w sposób *implicite*, tzn. bez konieczności wypełniania ankiet przez użytkownika.

Kolejnym problemem poruszonym w niniejszym rozdziale są problemy personalizacji systemów webowych. Według najprostszej definicji personalizacji mogą podlegać następujące elementy systemu webowego [Kobsa 2001]: zawartość, struktura i forma prezentacji. Dalsza część pracy niemal w całości poświęcona jest automatycznym metodom adaptacji tych elementów.

Przedstawiono perspektywy rozwoju systemów webowych Web 2.0 i Web 3.0. W systemach tego typu również można zastosować personalizację lub rekomendację interfejsów. W dalszej części pracy w punkcie 6.1 zaprezentowano aplikację typu Web 2.0 *Reporter*, opartą na wykorzystaniu mechanizmów Wiki do rekomendacji wiadomości oraz interfejsu użytkownika.



### 3. Modelowanie użytkownika w webowych systemach rekomendacyjnych

Obecnie zwiększa się liczba systemów webowych dostarczających swym docelowym użytkownikom informacji dostosowanej na ich potrzeby zarówno pod względem treści, jak i formy [Montaner 2003]. Systemy tego typu nazywane są systemami rekomendacyjnymi. Systemy rekomendacyjne wykorzystują wiele metod wypracowanych na gruncie wyszukiwania informacji i teorii agentów, a zwłaszcza agentów interfejsowych [Schneiderman 1997], które mogą być nawet utożsamiane z systemami rekomendacyjnymi.

Źródeł systemów rekomendacyjnych zarówno co do stosowanych metod, jak i dziedzin zastosowania można szukać przede wszystkim w systemach wyszukiwania informacji. Większość bardziej współczesnych metod wyszukiwania, szczególnie takich, które wykorzystują sprzężenie zwrotne relewancji (ang. *relevance feedback*) [Salton 1983], można z powodzeniem zaliczyć do systemów rekomendacyjnych. Również i te bardziej klasyczne metody wyszukiwania znajdują zastosowanie w dziedzinie systemów rekomendacyjnych (ang. *Recommendation Systems* – RS), np. metody z wykorzystaniem teczurusa dostarczającego pewnych stereotypów organizacji wiedzy w określonej dziedzinie.

W dalszej kolejności rozwój dziedziny systemów rekomendacyjnych był realizowany w ramach systemów agendowych, a zwłaszcza agentów interfejsowych [Wooldridge 1995], których podstawowym zadaniem była pomoc użytkownikowi w jego powtarzalnych czynnościach, takich jak filtrowanie poczty elektronicznej czy konstrukcji harmonogramu spotkań.

Model użytkownika w każdym systemie interakcyjnym, również w systemie rekomendacyjnym, stanowi z jednej strony jego uproszczenie pozwalające lepiej zrozumieć proces interakcji [Holt 2002], a z drugiej stanowi podstawę opisu i predykcji potencjalnych zachowań użytkownika w pracy z określonym systemem, co umożliwiła rzeczywistą implementację tego systemu [Newman, 1996]. W pracy [Spolsky 2001] modelem użytkownika nazywany jest zestaw przyzwyczajęń w pracy z programami oraz wyobrażeń użytkownika o tym jak dany program będzie działał. Natomiast termin profil użytkownika bywa czasem używany zamiennie z terminem model użytkownika [Montaner 2003] i [Daniłowicz 1992], bądź rozumiany nieco bardziej zwięźle jako konkretna implementacja modelu użytkownika.

W niniejszym rozdziale przedstawiono najpierw dziedziny zastosowań systemów rekomendacyjnych. Następnie przedstawiono cele użytkowników systemów rekomen-



dacyjnych. W kolejnych punktach niniejszego rozdziału zostaną przedstawione problemy reprezentacji, tworzenia i aktualizacji profilu użytkownika.

### 3.1. Dziedziny zastosowań systemów rekomendacyjnych

Podstawowymi dziedzinami zastosowania systemów rekomendacyjnych są: wspomaganie wyszukiwania, filtrowanie informacji i wspomaganie przeglądania (ang. *browsing*) informacji.

**Wspomaganie wyszukiwania informacji** w sieci WWW jest realizowane z powodzeniem przez różnorodne wyszukiwarki i metawyszukiwarki internetowe. Wyszukiwanie informacji nierozłącznie jest związane z pojęciem relewancji wyszukanych elementów, czyli ich zgodnością z potrzebami informacyjnymi użytkownika, który realizował wyszukiwanie [Salton 1975]. Wyszukiwarki internetowe bazują na zastosowaniu złożonych technik wyszukiwania pełnotekstowego [Rijsbergen 1979] oraz odwołań do danej strony z innych stron, przy czym waga takiego odwołania jest tym większa, im większa jest waga strony, z której odbywa się odwołanie. W wyszukiwarce Google technologia ważenia stron i odsyłaczy nazwana została Page Rank [Google 2004]. Wyszukiwarki internetowe mogą, oprócz zawartości treściowej dokumentów i odsyłanych hiperłączy, określając rangę wyszukanych dokumentów, brać pod uwagę także oceny relewancji wyszukiwanych elementów nadawane przez użytkowników, tzn. zwiększając rangę dla tych dokumentów, które zostały wybrane do pobrania po realizacji podobnego zapytania wyszukiwawczego.

Systemy rekomendacyjne służą bardzo często wyszukiwaniu i rekomendacji informacji określonego typu, np. rekomendacji filmów kinowych (np. *Recommender* [Montaner 2003] czy *Movie* [Nguyen 2007]), czy też rekomendacji ofert kupna nieruchomości, jak system *The Apt Decision* [Shearin 2001], wspomaganie wyszukiwania ofert w biurach podróży [Lenar 2007], czy w końcu system *LifeStyle Finder* rekomendujący oferty zakupu towarów i wycieczek. Ta dziedzina zastosowań zdecydowanie przeważa wśród wszystkich zastosowań systemów rekomendacyjnych [Herlocker 2004]. Systemy prezentują zwykle pewną listę (uporządkowaną lub nieuporządkowaną) najbardziej relewantnych elementów, jednak w niektórych zastosowaniach użytkownikom zależy na znalezieniu wszystkich relewantnych elementów. Ma to szczególnie istotne znaczenie w przypadku zastosowań prawniczych.

**Filtrowanie informacji** w przypadku jej wyszukiwania oznacza najczęściej usunięcie z odpowiedzi na zadane pytanie takich dokumentów, które ze względu na zawarte treści uznaje się za niepożądane. Dostarczyciele treści umieszczają na swoich stronach takie słowa, często zupełnie niezwiązane z treścią danej strony, aby zostały one wyszukane w odpowiedzi na najczęściej zadawane przez internautów pytania, takie jak seks, sport czy gry komputerowe. Przykładami takich systemów są np. *WebSail* [Chen 2000].

Kolejnymi dziedzinami zastosowań systemów rekomendacyjnych są systemy filtrujące pocztę elektroniczną (np. *Re-Agent*) i sieci wiadomości (np. *ACR News* lub *News Dude* [Montaner 2003]). Ich głównym przeznaczeniem jest odfiltrowanie spamu i mniej

interesujących wiadomości na podstawie dokonanej analizy wcześniej podejmowanych przez użytkownika akcji. W pracy [Herlocker 2004] niniejsza dziedzina zastosowań określana jest jako anotacja w kontekście i jej zadaniem jest określenie treści, dokumentów lub wiadomości, które warto zapoznać, od tych, które należy odfiltrować.

**Wspomaganie przeglądania** realizowane jest głównie przez systemy rekomendacyjne wspomagające wybór odsyłaczy hipertekstowych. System zwykle śledzi wybór odsyłaczy na odwiedzanych przez użytkownika stronach, a następnie rekomenduje podobne odsyłacze na nowych stronach. Jednym z najbardziej znanych przykładów tego typu systemów jest *Letizia* [Lieberman 1997], autonomiczny agent, który zapisuje URL odwiedzanych przez użytkownika stron, tworząc w ten sposób profil użytkownika, następnie używając prostej miary częstotliwości występowania słów kluczowych poszukuje sąsiedztwa odwiedzanej właśnie strony.

Oprócz trzech podstawowych dziedzin, systemy rekomendacyjne można również stosować do personalizacji interfejsów użytkownika w obrębie różnorodnych systemów webowych [Sobecki 2004b] oraz adaptacji strategii nauczania dla uczestników kursów prowadzonych z wykorzystaniem metod komputerowego wspomaganie nauczania, np. przygotowanie do egzaminu na prawo jazdy [Sobecki 2003b]. Różne metody i systemy wykorzystywane do adaptacji interfejsów użytkownika przedstawiono w dalszej części pracy w rozdziałach 5 i 6.

## 3.2. Cele i zadania użytkownika w systemie rekomendacyjnym

W celu pełniejszego poznania zagadnień związanych z systemami rekomendacyjnymi, oprócz zaprezentowanych różnych dziedzin ich zastosowania warto również zwrócić uwagę na cele i zadania użytkownika końcowego (ang. *end-user*) tych systemów [Herlocker 2004]. Perspektywa użytkownika jest szczególnie istotna do oceny użyteczności i dostępności interfejsu użytkownika. Cele użytkownika w oczywisty sposób realizowane są w ramach różnych zastosowań tych systemów wymienionych w punkcie 3.1. Jednak oprócz głównych celów, związanych z dziedzinami zastosowań, można wymienić wiele innych celów realizowanych przez samych użytkowników, które o ile nie zawsze są sprzeczne z celami stawianymi przez dysponentów tych systemów, o tyle z pewnością są dla nich o wiele mniej istotne bądź przynajmniej poboczne. Wśród nich można wymienić: surfowanie po sieci, weryfikację jakości rekomendacji, tuningowanie własnego profilu, wyrażenie siebie, pomoc innym oraz wywieranie wpływu na innych użytkowników.

**Surfowanie po sieci** jest częstą rozrywką wielu użytkowników Internetu, szczególnie atrakcyjnych dla nowych, mniej doświadczonych użytkowników webu, a tych wraz z rozwojem cywilizacji ciągle przybywa, szczególnie w krajach, w których odnotowuje się w ostatnich latach szybki rozwój gospodarczy oraz mających spory potencjał demograficzny, takich jak Chiny, Indie czy Brazylia [Miniwatts 2008]. Wielu użytkowników systemów, w tym również systemów rekomendacyjnych, takich jak *MovieLens* czy *Amazon.com* odwiedza je bez zamiaru realizowania jakichkolwiek zakupów, a jedynie do

zapoznania się z tymi systemami bądź dla samej przyjemności surfowania. Można przyjąć [Herlocker 2004], że dla użytkowników realizujących takie właśnie cele bardziej istotne od mechanizmów rekomendacyjnych są: „użyteczność” interfejsu użytkownika (jego atrakcyjność i łatwość użycia) oraz udostępniana zawartość informacyjna.

Potrzeba **weryfikacji jakości rekomendacji** wynika z tego, iż szczególnie w początkowym okresie użytkowania danego systemu użytkownicy wykazują się w stosunku do niego ograniczonym zaufaniem. Po pierwsze, użytkownicy „bawią się” systemem, aby się zorientować, czy rekomendacje odpowiadają ich gustom. Po drugie, użytkownicy modyfikują swoje profile w celu sprawdzenia, jaki wpływ mają zmiany profilu na zmianę rekomendacji. Przy okazji dokonano spostrzeżenia [Herlocker 2004], iż systemy komercyjne często rekomendują takie elementy (np. filmy na dvd), których użytkownicy nie znają, co wpływa na ich ograniczone zaufanie do rekomendacji, gdyż nie są w stanie jej zweryfikować.

**Tuningowanie (poprawa) profilu** realizowane jest przez użytkowników dokonujących oceny kolejnych elementów w nadziei, że dzięki temu uzyskają wyższą jakość rekomendacji. Co prawda ocena rekomendowanych elementów jest najczęściej wykorzystywanym sposobem sprzężenia zwrotnego w takich systemach, można jednak podciągnąć pod niniejszy cel wszelkie akcje związane z dostarczeniem bardziej kompletnej informacji na własny temat, czyli wszelkich danych określanych w pracy [Kobsa 2001] jako dane użytkownikowe „dane użytkownika” i użytkowania.

Kolejnym celem, dla już znacznie mniej licznej grupy użytkowników, jest wykorzystanie systemu rekomendacyjnego jako medium **wyrażania siebie** przez dostarczenie ogromnej liczby ocen elementów. Dla przykładu, wśród użytkowników systemu *MovieLens* [Herlocker 2004] można znaleźć takich, którzy ocenili ponad 1000 filmów. Użytkownicy ci dostarczali swoich ocen nie w celu poprawy własnego profilu, a dla własnej przyjemności oraz uzyskania forum do wyrażenia siebie i swojej opinii. Użytkowników takich nazywa się również *power user*. W przypadku tego typu użytkowników istotnymi są następujące elementy: zachowanie anonimowości na pożądanym poziomie (zupełna anonimowość bądź przeciwnie rozreklamowanie siebie), poczucie wnoszenia pewnego wkładu we wspólne „dzieło” oraz ułatwienia w dostarczaniu ocen.

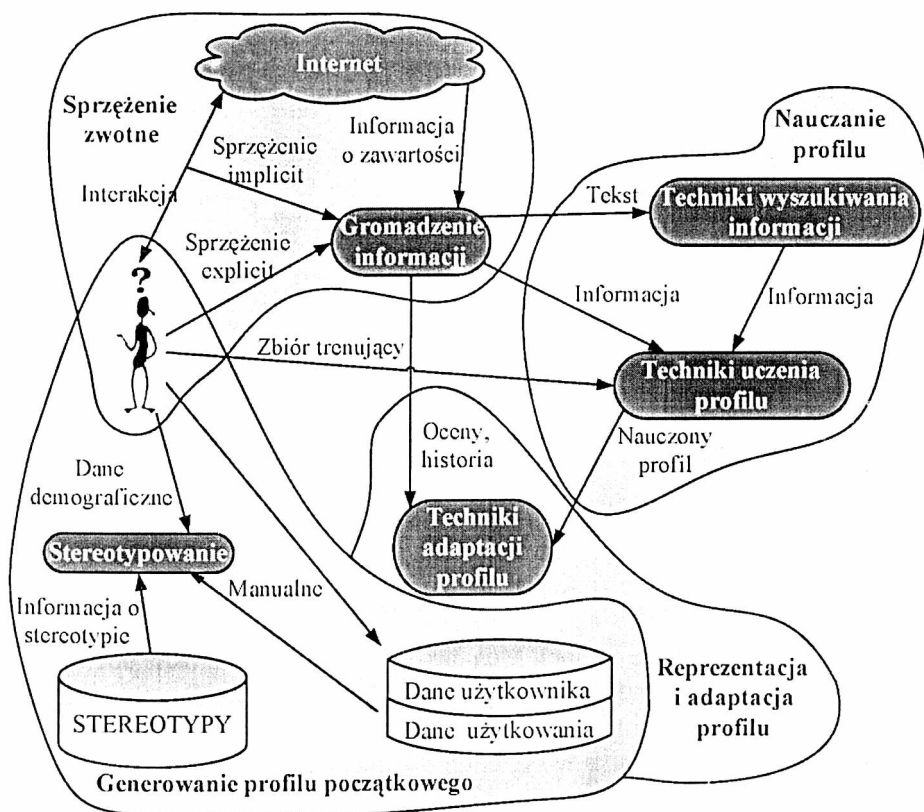
**Pomoc innym** jest kolejnym z celów dla określonej podgrupy użytkowników systemów rekomendacyjnych; co prawda może on być czasem realizowany równocześnie z wymienionym celem *power userów*, jednak często jest tak, że użytkownicy kierują się pobudkami czysto filantropijnymi, dostarczając wielu ocen jedynie z myślą o dobru wspólnym.

Za główny cel przyświecający bardzo wielu użytkownikom należy wymienić dążenie do **bezpośredniego wywierania wpływu na innych**. Dla przykładu, zwolennicy bądź nawet osoby zawodowo związane z określonym producentem filmowym wysoko oceniają określone filmy w systemie *MovieLens* [Herlocker 2004], nawet przed rozpoczęciem ich oficjalnej dystrybucji, aby zachęcić innych użytkowników do obejrzenia danego filmu. Oczywiście taka promocyjna działalność może nie leżeć w interesie ogółu użytkowników, dlatego systemy powinny zawierać mechanizmy, które spowodowałyby jej uniemożliwienie lub choćby ograniczenie.

### 3.3. Zawartość, reprezentacja, tworzenie i aktualizacja profilu użytkownika

Model użytkownika w systemie interakcyjnym stanowi z jednej strony uproszczenie, które pozwala na lepsze zrozumienie procesu interakcji zachodzącego w tym systemie [Holt 2002], a z drugiej stanowi podstawę zarówno opisu, jak i predykcji zachowań użytkownika w tym systemie. Model użytkownika pozwala również na rzeczywistą implementację systemu interakcyjnego [Newman, 1996]. Termin profil użytkownika bardzo często używany jest zamiennie z terminem model użytkownika [Kobsa 2001], przy czym profilowi przypisuje się zwykle nieco węższe znaczenie niż modelowi, ograniczając go zwykle do umożliwienia opisu i predykcji zachowań użytkownika w systemie.

Decyzje dotyczące zawartości profilu oraz jego reprezentacji stanowią pierwsze kroki na drodze budowy systemu rekomendacyjnego, gdyż od tych decyzji zależą wszelkie dalsze etapy związane z tworzeniem i utrzymaniem profilu w tego typu systemie [Montaner 2003]. Wzajemne zależności między różnymi elementami występującymi w procesie tworzenia, utrzymania i aktualizacji profilu użytkownika przedstawiono na rysunku 9, który jest modyfikacją rysunku przedstawionego w pracy [Montaner 2003].



Rys. 9. Generowanie i utrzymanie profilu

### 3.3.1. Zawartość profilu

Aby umożliwić projektantom i dysponentom informacyjnych systemów webowych dostosowanie ich na potrzeby użytkowników, należy opracować modele użytkowników. Podstawą do tworzenia modelu użytkownika jest zwykle, jak wynika z badań prowadzonych przez specjalistów z dziedziny interakcji człowiek–komputer (HCI) [Preece 1996], dokładne poznanie użytkownika, tzn. jego możliwości kognitywnych i uwarunkowań społecznych oraz w miarę dokładne poznanie jego potrzeb informacyjnych jak również przyzwyczajeń w interakcji. Ogólnodostępne webowe systemy informacyjne mogą mieć nawet miliony użytkowników, a ci różnią się między sobą pod wieloma względami; tak więc wypracowanie pojedynczego, uniwersalnego modelu odpowiedniego dla wszystkich jest niemożliwe [Sobecki 2001c]. Model użytkownika powinien zawierać dane dotyczące poszczególnych użytkowników i środowiska, w którym użytkownicy używają systemów webowych. Dane te mogą dotyczyć użytkownika, i zwane są wtedy danymi użytkownika (ang. *user data*), jak i jego interakcji z różnymi systemami webowymi, i zwane są wtedy danymi użytkowania (ang. *usage data*), natomiast dane dotyczące platform sprzętowych i systemowych oraz środowiska, w których systemy te są wykorzystywane nazywane są danymi środowiskowymi (ang. *environment data*) [Kobsa 2001].

#### 3.3.1.1. Dane użytkownika

Dane użytkownika zawierają jego osobistą charakterystykę [Kobsa 2001]. Dane użytkownika można podzielić na następujące kategorie: dane demograficzne „dane demograficzne”, wiedza użytkownika, umiejętności użytkownika, zainteresowania i preferencje użytkownika oraz cele i plany użytkownika.

Kobsa w swojej pracy [Kobsa 2001] dzieli dane demograficzne na następujące kategorie: dane zapisowe (np. nazwisko, adres, nr telefonu), dane geograficzne (np. kod pocztowy, miasto, stan, kraj), charakterystyka użytkownika (np. wiek, płeć, wykształcenie, dochód itp.), dane psychograficzne (np. określające styl życia), zachowania konsumenckie (np. częstość użycia produktu) oraz dane o rejestracji w różnego rodzaju serwisach. Dane te szczególnie skojarzone z wysokiej jakości danymi stereotypowymi mogą przyczynić się do wzrostu jakości rekomendacji.

Wiedza użytkownika obejmuje dane dotyczące założeń o wiedzy, lub ogólniej, przekonaniach, o pojęciach, relacjach między nimi, oraz dane dotyczące faktów i reguł, które dotyczą dziedziny aplikacji systemu rekomendacyjnego [Kobsa 2001]. Dane tego typu bardzo często wykorzystywane są do personalizacji prezentowanej informacji. Dla przykładu, w systemie badawczym *Megane* promującym nowo wprowadzany ówczesnie model samochodu osobowego Renault Megane II [Sobecki 2004b] dane dotyczące wiedzy użytkownika z dziedziny motoryzacji miały wpływ na szczegółowość i słownictwo prezentowanej informacji.

Do modelowania użytkownika w systemie rekomendacyjnym, oprócz danych dotyczących wiedzy użytkownika, mogą również ze sporym powodzeniem zostać wykorzystane dane dotyczące jego umiejętności i możliwości użytkownika. Dla przykładu, w systemie *Unix Consultant* w zależności od posiadanego przez użytkownika stopnia

znajomości systemu Unix dostosowuje się odpowiednie komunikaty i treści pomocy [Kobsa 2001]. W innym systemie, *AVANTI* [Fink 1998], przeznaczonym do dostarczenia informacji turystycznej, w zależności od rodzaju niepełnosprawności użytkownika prezentowane są informacje dotyczące lokacji, które dany użytkownik może wygodnie odwiedzić.

Kolejna grupa danych użytkownika obejmuje jego zainteresowania i preferencje, które mogą w sposób niemal bezpośredni zostać wykorzystane w rekomendacji, np. w drodze mało skomplikowanego wnioskowania stereotypowego. Dla przykładu, w systemie rekomendującym przepisy kulinarne *Kucharek* [Sobecki 2007], zadeklarowanie przez użytkownika, iż preferuje kuchnię jarską spowoduje, że przepisy na dania zawierające mięso nie będą prezentowane.

Ostatnia grupa danych użytkownika obejmuje cele i plany użytkownika, do których należą szczegółowiej omówione w punktach 3.1 i 3.2: nawigacja, wyszukiwanie informacji, zakup towarów, pomoc innym i wyrażanie siebie. System rekomendacyjny powinien się adaptować w sposób ułatwiający użytkownikom realizację ich celów. Problemem może być jednak samo rozpoznanie tych celów. Można przyjąć, że użytkownik sam je wyspecyfikuje, tak jak np. w systemie *HYPERFLEX*, system zaś zaprezentuje ścieżki postępowania wcześniejszych użytkowników, którzy określili ten sam cel [Kobsa 2001].

### 3.3.1.2. Dane użytkowania

Dane użytkowania zawierają zapisy dotyczące różnych aspektów korzystania przez konkretnego użytkownika z różnych systemów webowych. W zależności od stosowanych technologii istnieje spore zróżnicowanie w możliwościach pozyskiwania danych. Korzystanie z aplikacji opartych na protokole http i prostych stronach HTML pozwala jedynie na uzyskanie informacji dotyczących odwiedzanych stron w danym serwisie, ale już korzystanie z systemów opartych na takich technologiach, jak *cookies*, DoubleClick, Java, JavaScript czy actiontracking umożliwia uzyskanie wielu dodatkowych danych dotyczących aktywności danego użytkownika w innych serwisach, bądź dokładniejszych danych o interakcji użytkownika z danym systemem webowym (np. położenie kursora w określonym czasie).

Dane użytkowania wiążą się zatem ze sposobami zbierania danych podczas interakcji użytkownika z systemem, które zostaną opisane dokładniej w punkcie 3.3.3. Dane użytkowania obejmują dwie główne kategorie [Kobsa 2001]: dane obserwowalne i regularności użycia. Dane obserwowalne można podzielić na: akcje selekcji, czas oglądania strony, ocenianie, zakupy i akcje z nimi związane, oraz inne akcje potwierdzające. Regularność użycia obejmuje natomiast: częstość użycia, korelacje sytuacja–akcja i sekwencje akcji.

**Akcje selekcji** realizowane są w systemach webowych przez kliknięcie odsyłacza hipertekstowego. Użytkownik dokonuje wyboru spośród umieszczonych na stronie odsyłaczy i realizuje wybór jednego z prezentowanych rekomendowanych elementów (np. filmu, książki czy innego produktu). Akcje selekcji mogą być indykatorami kilku typów danych użytkownika (patrz punkt 3.3.1.1), takich jak: zainteresowania użytkownika (uży-



kownik otwiera strony webowe kierując się swoimi zainteresowaniami, jak w przypadku systemu *Letizia* [Liebermann 1995]), niewiedza w jakimś aspekcie (użytkownicy często klikają na odsyłacze prowadzące do szerszych objaśnień nieznanymi pojęć, np. technicznych) oraz gusta (użytkownicy systemów rekomendacyjnych wskazują na swoje preferencje podczas procesu oceniania wybranych elementów, np. filmów czy utworów muzycznych) [Sobecki 2006].

**Czas oglądania** strony może przynieść wiele informacji dotyczących użytkownika, o ile jego pomiar będzie miarodajny, gdyż często bywa tak, że strona może być otwarta, a użytkownik poświęca swoją uwagę zupełnie innej stronie webowej w innym otwartym oknie. Pomiary mogą być dokładniejsze, gdy zastosuje się metody actiontrackingu [Zatuchin 2008]. Bardzo krótki czas oglądania może stanowić podstawę negatywnej informacji o zainteresowaniach użytkownika.

Kolejnym istotnym elementem danych obserwowalnych jest **ocenianie**, tzn. nadawanie explicite ocen prezentowanym elementom. Oceny takie w sposób bezpośredni wskazują relewancję [Salton 1983] tych elementów dla użytkownika. W systemach rekomendacyjnych wykorzystuje się bądź oceny binarne (np. relewancjne i nirelewancjne), bądź też ograniczoną skalę dyskretną, np. liczbową od 1 do 5 w przypadku systemu *MovieLens* [Good 1999] albo opisową, jak w przypadku systemu *Syskill & Webert* [Pazzani 1996], gdzie wyróżniono trzy poziomy oceny: gorący (ang. *hot*), letni (ang. *lukewarm*) oraz zimny (ang. *cold*). Używanie takich ocen może powodować pewne problemy. Po pierwsze, użytkownik, dokonując oceny danego elementu, kieruje się subiektywnymi odczuciami, w dużej mierze zależnymi od konkretnej sytuacji i czasu, w którym nadawana jest ocena; tymczasem projektanci systemów spodziewają się raczej ocen obiektywnych. Drugim problemem jest fakt, że użytkownicy niechętnie dostarczają jakichkolwiek ocen, gdyż nie są one najczęściej postrzegane jako prowadzące bezpośrednio do celu, jaki sobie użytkownicy stawiają. Zauważono również, że przyznawanym przez użytkowników ocenom często nie można ufać [Kobsa 2001], co metody akwizycji danych powinny brać pod uwagę, ze szczególnym uwzględnieniem ocen negatywnych [Schwab 2000].

Bardzo istotną rolę w modelowaniu użytkowników w systemach rekomendacyjnych pełnią obecnie **zakupy i wszelkie akcje z nimi związane**, gdyż uznawane są one za mocny indyktor zainteresowań klienta. W modelu użytkownika można uwzględnić cechy nabytych produktów (np. autor książki) celem modelowania przyszłych potrzeb klienta. Użytkownicy zwykle dokonują zakupów na swoje własne potrzeby, dlatego informacja o dokonanych zakupach może stanowić podstawę do klasyfikowania lub grupowania użytkowników. Zdarza się jednak, że użytkownicy dokonują przez Internet zakupów nie na swoje potrzeby, ale jako prezent dla członków swojej rodziny lub swoich przyjaciół. W takiej sytuacji zakupy nie charakteryzują preferencji samych kupujących, a raczej obdarowywanych. Możliwym rozwiązaniem służącym oddzieleniu takich zakupów jest wyłączenie transakcji dokonywanych w okresach nasilonych zakupów prezentów, m.in. Bożego Narodzenia, Walentynki czy innych tego typu świąt. Najbardziej popularnym systemem o światowym zasięgu wykorzystującym do modelowania cechy zakupionych (lub w inny sposób wskazanych) produktów jest *Amazon.com*. System ten



rozwiązuje problemy zakupów prezentów dla osób trzecich w drodze ignorowania w procesie analizy tych produktów, których adres dostarczenia jest różny od adresu klienta.

Oprócz akcji potwierdzających lub niepotwierdzających można również wymienić **inne akcje potwierdzające**, takie jak: zapisanie dokumentu, jego drukowanie, przeniesienie adresu do zakładek lub jego usunięcie z zakładek [Kobsa 2001]. Tego typu akcje, chociaż nie mają same w sobie dużej mocy indykatywnej, mogą jednak wzmacniać lub osłabiać pewne elementy modelu użytkownika.

Drugim typem głównym w obrębie danych użytkownika są **regularności użycia**, które co prawda nie prowadzą bezpośrednio do adaptacji to jednak wskazują na moc zainteresowań i nawyków użytkownika. Wśród regularności użycia wyróżnia się: częstotliwość użycia, korelacje sytuacji i akcji oraz sekwencje akcji. Jako przykład wykorzystania częstotliwości użycia oraz sekwencji akcji można przytoczyć np. Asystenta pakietu MS Office, który w przypadku często powtarzanych sekwencji akcji proponuje ich zapisanie w postaci makroinstrukcji. Korelacje sytuacji i akcji wykorzystywane są np. w agentach interfejsowych [Brown 1998], których podstawą działania jest modelowanie przyszłych zachowań na podstawie przeszłych zachowań użytkowników, np. otwieranie lub kasowanie bez czytania określonych wiadomości email.

### 3.3.1.3. Dane środowiskowe

Dane środowiskowe, czyli zarówno platformy softwarowe i sprzętowe, jak i miejsce oraz okoliczności związane z ich wykorzystaniem, stanowią oprócz danych użytkownika i danych użytkownika bardzo istotny element modelowania w systemach rekomendacyjnych, a zwłaszcza w tych systemach, które wykorzystują metody wnioskowania bazujące na przypadkach użycia (ang. *case-based reasoning* CBR). Dane środowiskowe powinny być brane pod uwagę również w przypadku projektowania standardowych systemów interakcyjnych [Newman 1996].

Środowisko softwarowe obejmuje zarówno dane dotyczące systemu operacyjnego, przeglądarki internetowej wykorzystywanej do komunikacji z systemem webowym, możliwości wykorzystania w tej przeglądarce takich mechanizmów, jak JavaScript czy Java, oraz zainstalowanych pluginów (np. Flash czy RealMedia).

Środowisko sprzętowe jest obecnie bardzo bogate, gdyż obejmuje nie tylko komputery klasy PC, stacje robocze, ale coraz częściej wykorzystywane notebooki, jak i mobilne PDA czy nawet bardziej rozbudowane telefony komórkowe. Do środowiska sprzętowego zaliczane są również: przepustowość połączenia z Internetem, prędkość przetwarzania, możliwości wyświetlacza i zastosowane urządzenia wejściowe.

Dane dotyczące miejsca użycia mogą dotyczyć zarówno informacji wskazujących na położenie geograficzne a pochodzących np. z urządzenia GPS, jak i informacji dotyczących okoliczności towarzyszących użytkownikowi systemu, takich jak natężenie oświetlenia czy poziom hałasu.

### 3.3.2. Reprezentacja profilu użytkownika

W systemach rekomendacyjnych wykorzystywanych jest wiele różnych metod reprezentacji profilu użytkownika. W pracy [Montaner 2003] zostały wyróżnione nastę-

pujące typy reprezentacji: historia zakupów, nawigacji webowej i email; ważne wektory cech;  $n$ -gramy; sieci semantyczne; sieci asocjacyjne; klasyfikatory obejmujące sieci neuronowe, drzewa decyzyjne, reguły indukcyjne lub sieci bayesowskie; macierz ocen i zbiory cech demograficznych. Niniejszy zestaw można rozszerzyć o inne metody reprezentacji, takie jak krotki [Nguyen 2003], [Sobecki 2004] i struktury drzewiaste [Sobecki 2002] i [Sobecki 2003].

### 3.3.2.1. Historia: zakupów, nawigacji webowej i emaili

Jednym z najprostszych, a przy tym bardzo przydatnym sposobem reprezentacji profilu, jest rejestracja różnorodnych akcji użytkownika dokonywanych w trakcie interakcji z systemem. Można tu wymienić historie (listy): dokonywanych przez użytkownika zakupów w sieci, odwiedzanych przez użytkownika stron webowych oraz przeczytanych czy wysłanych emaili.

**Lista zakupów** najczęściej wskazuje na preferencje konsumenckie użytkowników systemów webowych. Podczas ich stosowania należy jednak pamiętać o ich zaletach i wadach, które zostały wymienione w punkcie 3.3.1.2 w ramach opisanego zakupów i wszelkich akcji z nim związanych.

Kolejnym typem reprezentacji profilu jest **historia nawigacji** w sieci webowej w postaci listy adresów URL odwiedzonych przez użytkownika stron. Przeglądarki internetowe (takie jak IE, Netscape i Opera) przechowują wszystkie adresy URL odwiedzonych stron (np. w IE w pewnym okresie), co powoduje, że podczas przeglądania dowolnej strony WWW przeglądarka wyświetla odsyłacze hipertekstowe odwiedzonych stron w inny sposób niż odsyłacze do stron do tej pory nieodwiedzonych. Różnica ta standardowo przedstawiana jest jako różnica w kolorze: niebieski dla stron nieodwiedzonych i fioletowy dla odwiedzonych. Pomimo że sam URL nie niesie ze sobą zbyt wielu informacji (czasami bardziej informatywna może być nazwa pliku lub ścieżka dostępu [Sobecki 1998]), to w połączeniu z wiedzą odnośnie do zawartości wskazywanej strony może stanowić przydatny do modelowania użytkownika element. Problemy z gromadzeniem i przekazywaniem listy odwiedzonych stron będą poruszone w sekcji dotyczącej tworzenia i utrzymania profilu użytkownika.

Ostatnim typem reprezentacji profilu jest **lista przeczytanych lub wysłanych emaili** i wiadomości newsowych. Pomaga ona identyfikować osoby, z którymi użytkownik koresponduje oraz tematy jego zainteresowań, co może dostarczać podstaw do rekomendowania porządkowania i filtrowania wiadomości.

### 3.3.2.2. Wektory: boolowskie, binarne, cech i ważne wektory cech

W modelu wektorowym profile reprezentowane są jako wektor cech, zwykle terminów bądź pojęć powiązanych z pewną wartością, która może być boolowska, binarna lub rzeczywista i w zależności od typu tej wartości rozróżniane są odpowiednie typy wektorów. W wektorze boolowskim powiązana wartość reprezentuje obecność – przyjmując wartość prawdy bądź nieobecność – przyjmując wartość fałszu, określonej cechy w profilu [Montaner 2003]. Wektor boolowski może być reprezentowany przez wektor binarny, w którym wykorzystuje się wartości 0 lub 1.

Wektor binarny był bardzo często wykorzystywany w różnego rodzaju zastosowaniach z dziedziny wyszukiwania informacji (IR) w celu reprezentacji poszczególnych dokumentów lub zapytań informacyjnych [Salton 1983]. Wektor binarny  $\mathbf{v}$  o wymiarze  $n$  można zdefiniować w następujący sposób:  $\mathbf{v} = (f_1, f_2, \dots, f_n)$ , gdzie  $f_i \in \{0, 1\}$  oraz  $i = 1, 2, \dots, n$ . W dziedzinie IR każda wartość  $f_i$  oznacza, że określony term  $i$  jest obecny ( $f_i = 1$ ) lub nieobecny ( $f_i = 0$ ) w dokumencie lub zapytaniu.

Wektor cech  $\mathbf{v}$  o wymiarze  $n$  zdefiniowany jest następująco:  $\mathbf{v} = (f_1, f_2, \dots, f_n)$ , gdzie  $f_i$  jest wartością wyrażoną liczbą rzeczywistą pewnej cechy  $i$ , gdzie  $i = 1, 2, \dots, n$ . W pracy [Basili 2003] przedstawiono następujące własności reprezentacji w postaci wektora cech:

- nie zakłada się żadnych formalnych relacji między cechami (niezależność *a priori*),
- dopuszcza się wyłącznie jedną wartość każdej cechy (pewność obserwacji).

Ostatnim typem wektora jest ważony wektor cech, który charakteryzuje się zastosowaniem wag modyfikujących wartości w zwykłym wektorze cech, np. niektóre termy albo atrybuty mogą mieć większą lub mniejszą wagę w całym wektorze. Wazony wektor cech  $\mathbf{v}$  o wymiarze  $n$  i wagach  $\mathbf{w} = (w_1, w_2, \dots, w_n)$  przypisanych dla każdej z  $n$  cech, można więc zdefiniować w następujący sposób:  $\mathbf{v} = (w_1 f_1, w_2 f_2, \dots, w_n f_n)$ , gdzie  $f_i$  jest wartością wyrażoną liczbą rzeczywistą pewnej cechy  $i$ ,  $w_i$  wagą tej cechy również wyrażoną wartością rzeczywistą dla każdego  $i \in \{1, 2, \dots, n\}$ .

Wektory są bardzo popularną metodą reprezentacji profili użytkownika w systemach rekomendacyjnych. Według pracy [Montaner 2003] wazone wektory cech są najpopularniejszą metodą reprezentacji profili pośród wszystkich analizowanych systemów rekomendacyjnych, należą do nich m.in. *Letizia* [Liebermann 1997], *MovieLens* [Good 1999] i *WebMate* [Chen 1998]. Reprezentację w postaci wektorów boolowskich wykorzystano w systemach *WebSail* i *WebWatcher*.

### 3.3.2.3. Krotki

Profil użytkownika w systemie rekomendacyjnym może być reprezentowany przez krotkę, którą można zdefiniować w następujący sposób: niech  $A$  oznacza skończony zbiór atrybutów, a  $V$  zbiór wartości atrybutów, gdzie  $V = \bigcup_{a \in A} V_a$  ( $V_a$  jest dziedziną atrybutu  $a$ ), krotka jest funkcją  $t: A \rightarrow V$ , gdzie  $\forall_{a \in A} (t(a) \in V_a)$ . Ten typ reprezentacji można nazwać za Pawlakiem [Pawlak 1983] jednowartościowym systemem informacyjnym.

W zdefiniowanych krotkach, wartości atrybutów są atomowe. Można jednak rozważyć również inne rodzaje wartości, takie jak zbiory, zbiory uporządkowane lub zbiory uporządkowane z powtórzeniami. Aby zdefiniować tego typu krotki, należy wprowadzić pojęcie  $\Pi(V_a)$ , które określa zbiór podzbiorów  $V_a$ , jednocześnie można określić  $\Pi(V)$ , gdzie  $\Pi(V) = \bigcup_{a \in A} \Pi(V_a)$ . Wtedy krotkę  $t$  można zdefiniować jako funkcję  $t: A \rightarrow \Pi(V)$ , gdzie  $\forall_{a \in A} (t(a) \in \Pi(V_a))$ .

Krotki były również wykorzystywane w systemach rekomendacyjnych do modelowania profili użytkownika do adaptacyjnych interfejsów użytkownika opisanych np. w pracach [Nguyen 2003a], [Sobecki 2004b]. Wyniki tych prac zostały również zaprezentowane w niniejszej pracy w rozdziale 5, gdzie można znaleźć przykład zastosowań krotek do reprezentacji profilu użytkownika.

### 3.3.2.4. Inne metody reprezentacji profili

Przedstawione metody reprezentacji profili użytkownika w systemach rekomendacyjnych należą z jednej strony do dość prostych, a z drugiej strony do bardzo często wykorzystywanych metod. Wśród zrealizowanych dotychczas implementacji systemów rekomendacyjnych można znaleźć wiele innych i to bardziej złożonych metod reprezentacji [Montaner 2003]. Należą do nich następujące metody: ważne  $n$ -gramy, ważne sieci semantyczne, ważne sieci asocjacyjne, oceny użytkowników, dane demograficzne oraz metody bazujące na klasyfikatorach. Do innych metod reprezentacji profili można zaliczyć również struktury drzewiaste [Sobecki 2002b], dokumenty XML [Sobecki 2003] oraz struktury sieciowe [Sobecki 2008a], które szczegółowo zostaną przedstawione w punkcie 6.2.1.

**Ważne  $n$ -gramy** umożliwiają reprezentację profilu w postaci sieci słów z wagami przypisanymi zarówno do węzłów jak i krawędzi sieci. Wagi te reprezentują współwystępowanie połączonych krawędzią słów.

**Ważne sieci semantyczne** przeznaczone są do reprezentowania informacji o profilach użytkowników przez zastosowanie sieci połączonych ze sobą słów czy pojęć [Collins 1969]. Połączenia zwane w sieci semantycznej łukami odzwierciedlają pewne relacje binarne między węzłami, takie jak całość–część i klasa–superklasa. W systemach rekomendacyjnych *ifWeb* i *SiteIF* pojęcia reprezentowane przez sieć semantyczną dodatkowo uzupełnione są o pewne wagi, które odzwierciedlają zainteresowanie użytkownika określonym tematem wyrażonym słowami. Natomiast **ważne sieci asocjacyjne**, w odróżnieniu od sieci semantycznych, nie rozróżniają typów relacji łączących ze sobą słowa w sieci.

**Modele bazujące na klasyfikatorach** obejmują: sieci neuronowe, drzewa decyzyjne i sieci bayesowskie. Modele te łączą w sobie zarówno metodę reprezentacji, jak i wykorzystania zgromadzonej wiedzy, dlatego zostaną nieco szerzej omówione w dalszej części pracy.

**Dane demograficzne** stanowią pewnego rodzaju specjalizacją zaprezentowanych metod reprezentacji, takich jak wektor cech i krotki. Charakterystyka danych demograficznych zostanie przedstawiona w dalszej części pracy.

**Struktury drzewiaste** [Sobecki 2002b] umożliwiają w odróżnieniu od innych metod reprezentacji, takich jak wektor cech i krotki, na uwzględnienie hierarchicznych zależności między poszczególnymi atrybutami opisującymi cechy. Struktury tego typu bardzo dobrze nadają się do modelowania rekomendowanych elementów, np. interfejsów użytkowników, które często stanowią ważny element całego modelu użytkownika. W pracy [Sobecki 2002b] ten typ reprezentacji wyrażony jest przez strukturę obejmującą zbiory węzłów, krawędzi, węzła będącego korzeniem, zbiory atrybutów i ich wartości oraz trzech funkcji, które przyporządkowują odpowiednio węzłom atrybuty, atrybutom ich wartości oraz funkcja odległości między poszczególnymi wartościami określonych atrybutów. Zależności drzewiaste można również reprezentować w postaci dokumentu XML [Sobecki 2003], co ma tę zaletę, że są proste w implementacji i wygodne do przetwarzania z wykorzystaniem wielu gotowych narzędzi programistycznych.

### 3.3.3. Inicjowanie profilu

Profil użytkownika w momencie, gdy zaczyna on pracę z systemem, może mieć różną zawartość: pustą, wprowadzoną manualnie, modelowaną na podstawie stereotypów oraz utworzoną na podstawie zestawu uczącego.

W wielu systemach rekomendacyjnych nie stawia się wymagania, aby na rozpoczęciu pracy profil zawierał jakiegokolwiek dane. Systemy te zakładają, że z początku profil użytkownika jest **pusty** i wypełnia się danymi w miarę, jak użytkownik pracuje z systemem, np. lista zakupów jest rozszerzana w miarę nabywania w sklepie internetowym kolejnych towarów, a lista adresów URL rozszerzana jest w miarę odwiedzania kolejnych stron WWW.

Innym bardzo często stosowanym w systemach rekomendacyjnych rozwiązaniem dotyczącym inicjacji zawartości profilu użytkownika jest **manualne** bezpośrednie opisanie oczekiwań rekomendacyjnych przez samego użytkownika. Użytkownik realizuje to zadanie zwykle przez wpisanie odpowiednich słów kluczowych do pewnego formularza bądź wybranie określonych kategorii z przygotowanego zestawu.

Manualne inicjowanie profilu ma podstawową zaletę wynikającą z tego, że po wprowadzeniu określonych oczekiwań co do rekomendacji użytkownik jest świadomy dlaczego dane elementy są rekomendowane. Rozwiązanie to ma jednak pewne wady wynikające przede wszystkim z trudności w określeniu własnych oczekiwań oraz z dużego nakładu pracy użytkownika związanego z określeniem zainteresowań użytkownika [Montaner 2003].

Dane wprowadzane są przez użytkownika np. w drodze wypełniania formularza dostarczonego przez system webowy i dotyczą najczęściej danych demograficznych. Formularze mogą zawierać różnorodne pola identyfikujące użytkownika. Lista najczęściej spotykanych atrybutów, które użytkownik ma określić w formularzu obejmuje (na przykładzie serwisów Wirtualna Polska i Lycos): imię, nazwisko, płeć, datę urodzenia i/lub wiek, dane adresowe wraz z kodem pocztowym i krajem, dane kontaktowe (nr telefonu i adres email), zawód, wykształcenie, zainteresowania.

W przypadku podawania danych bezpośrednio przez użytkownika można spodziewać się pewnych problemów [Kobsa 2002]. Przede wszystkim użytkownicy obawiają się, i to bardzo często nie bezpodstawnie, utraty swojej prywatności, a w rezultacie rezygnują z rejestracji lub podają nieprawdziwe dane. Ogólnie troska o kwestie bezpieczeństwa w transakcjach e-commerce [TNS 2002] jest np. dla prawie 75% Czechów i Niemców głównym powodem niepodawania numerów kart kredytowych.

Wykorzystanie **stereotypu** polega na zaklasyfikowaniu użytkownika do określonej kategorii, co skutkuje najczęściej wyborem odpowiedniej rekomendacji, np. umożliwia odpowiedni dla danego użytkownika dobór reklam czy też selekcję lub prezentację wiadomości, np. poczynając od tych, które należą do kategorii jego zainteresowań bądź też związane są z jego miejscem zamieszkania (np. prognoza pogody podawana dla określonego regionu). Klasyfikacja może odbywać się przez wykorzystanie danych wprowadzonych przez użytkownika do odpowiedniego formularza lub za pomocą identyfikacji użytkownika przez zapisy umieszczone w plikach *cookies*.

Wypełnianie formularza może być zarówno opcjonalne, jak i obligatoryjnie przed rozpoczęciem pracy z systemem bądź dostępem do pewnych wybranych opcji. Bardzo często tylko niektóre pola formularza, takie jak login czy kod pocztowy są wymagane, a pozostałe, takie jak zainteresowania czy zawód są uznaniowe. Dokładna zawartość formularzy została omówiona podczas prezentacji danych użytkownika w punkcie 3.3.1.1.

Na podstawie dodatkowych badań statystycznych oraz informacji zebranych np. podczas spisów powszechnych organizowanych w wielu krajach co kilka lat można wydobyc z tego typu danych o wiele więcej informacji. Już w latach 60. ubiegłego stulecia, Jonathan Robbin, amerykański specjalista w dziedzinie marketingu zauważył, że kod pocztowy związany z każdym miejscem zamieszkania w Stanach Zjednoczonych może służyć jako bardzo dobry wskaźnik statusu społecznego, preferencji politycznych, zainteresowań i wielu innych zachowań marketingowych. Zauważył on mianowicie, że ludzie mieszkający na tej samej ulicy kupują podobne samochody oraz oglądają te same programy telewizyjne [Quinn 2001]. Obecnie prace te są krytykowane, przede wszystkim ze względu na systemy przydzielania tych samych kodów pocztowych nawet bardzo długim arteriom bądź stronom ulicy, które to mają bardzo zróżnicowany status materialny swoich mieszkańców. Podobny problem występuje także w Polsce, gdzie bardzo często całe miasta, nawet powiatowe i inne miejscowości, mają ten sam kod pocztowy, przykładem może tu być miasto powiatowe Lubliniec w województwie śląskim, mające kod pocztowy taki sam jak okoliczne wioski.

Dane tego typu mają oczywiście charakter mocno przybliżony, jednak biorąc pod uwagę, że systemy webowe wykorzystywane są codziennie nawet przez miliony użytkowników, tego rodzaju statystyczne podejście jest jak najbardziej uzasadnione.

Inną metodą realizacji inicjowania profilu na podstawie stereotypów jest wykorzystanie mechanizmu cookies (omówionego w punkcie 2.3.1.2). Mechanizm ten pozwala zidentyfikować użytkownika i przypisać mu pewną historię jego zachowań, w postaci odwiedzonych stron, oglądanych reklam, zakupionych towarów lub nawet wypełnionych formularzy.

**Zbiór uczący** wykorzystywany jest w wielu metodach sztucznej inteligencji, np. w sieciach neuronowych, gdzie jest on stosowany w fazie przygotowującej każdą sieć do pracy [Osowski 2000]. W systemach rekomendacyjnych wykorzystanie zestawu uczącego do zainicjowania profilu użytkownika polega na zaprezentowaniu użytkownikowi pewnej sekwencji elementów podlegających rekomendacji wraz z prośbą o ocenę ich relewancji względem potrzeb użytkownika. Dla przykładu w eksperymentalnym internetowym sklepie muzycznym [Sobecki 2006], użytkownik wybiera lubiane i nielubiane utwory, co pozwala zakwalifikować użytkownika do kategorii fana rocka, heavy-metalu czy też trash-metalu.

Zaletą zbioru uczącego jest bardzo prosta implementacja, natomiast wadą jest zagrożenie, iż dobór tego zbioru nie będzie wystarczająco reprezentatywny i selektywny [Montaner 2003]. W przytoczonym przykładzie wykorzystania zbioru uczącego do klasyfikacji preferencji muzycznych zastosowanie tzw. wielkich klasyków muzyki rockowej, takich jak *Smoke on the water* czy *Stairway to heaven* może nam niewiele powie-



dzień o preferencjach użytkownika, gdyż tego typu utwory podobają się większości klientów sklepów muzycznych bez względu na specyfikę ich upodobań.

### 3.3.4. Techniki uczenia profilu

Uczenie profilu (ang. *profile learning*) występuje w przypadku gdy podczas tworzenia profilu wykorzystuje się pewne dane użytkownika, zdobyte np. z użyciem zbioru uczącego lub zebrane podczas użytkowania systemu przez użytkownika. Wykorzystanie zbioru uczącego często stanowi krok wstępny wymagany do pełnego wykorzystania systemu. Jednak użytkownik w czasie pracy z systemem ma możliwość dokonywania zmian swoich preferencji, a także ma możliwość dodawania danych uzupełniających pozwalających na poprawę efektywności rekomendacji dostarczanych przez system.

Na etapie uczenia profilu dokonywana jest również strukturalizacja danych, polegająca przede wszystkim na ekstrakcji informacji z nieustrukturalizowanych danych, np. w postaci obejrzanych stron internetowych zapisywanych w postaci profilu według wybranej metody jego reprezentacji.

Techniki uczenia profilu mogą działać zarówno on-line, jak i off-line. Istnieją systemy, które wykorzystują obydwie te techniki, jak również i takie, które poprzestają na wykorzystaniu jednej z nich. W przypadku zastosowania wyłącznie techniki on-line mamy do czynienia z tzw. leniwym uczeniem (ang. *lazy learning*).

Techniki off-line umożliwiają znalezienie modelu zachowań użytkownika, natomiast techniki on-line – na bieżąco go aktualizować. Mają one zastosowanie w sytuacji, gdy preferencje użytkowników zmieniają się dość wolno w stosunku do czasu tworzenia modelu [Montaner 2003]. W sytuacji dynamicznych zmian preferencji użytkownika, technika off-line jest nieodpowiednia.

W taksonomii agentów rekomendacyjnych [Montaner 2003] wyróżniono następujące cztery rodzaje rozwiązania problemu wykorzystania technik uczenia profilu: brak zastosowania technik uczących, strukturalne techniki wykorzystywane w dziedzinie IR, klasyfikacja oraz grupowanie.

W istniejących systemach rekomendacyjnych nie stosuje się żadnej z technik uczenia profilu wówczas, gdy uzyskiwane dane użytkownika wykorzystywane są bezpośrednio w takiej postaci, w jakiej są pozyskiwane przez system rekomendacyjny. Dzieje się tak najczęściej wtedy, gdy wykorzystywane są takie reprezentacje profili, jak listy zakupów stosowane w systemach rekomendacyjnych typu *Amazon.com* lub *CDNow*. Innym przypadkiem są systemy wykorzystujące takie techniki filtrowania, jak filtrowanie demograficzne i kolaboratywne, które zostaną przedstawione w punkcie 4.1.

#### 3.3.4.1. Techniki strukturalizacji wykorzystywane w dziedzinie IR

Podstawowym zadaniem techniki strukturalizacji jest ekstrakcja informacji z podanych na wejściu danych niestrukturalnych. Typowym przykładem danych niestrukturalnych są teksty w języku naturalnym, np. teksty odwiedzanych stron lub opisy wybranych elementów (np. filmów). W ramach dziedziny wyszukiwania informacji (IR) opracowano wiele metod strukturalizacji, w ramach której można wyróżnić dwa podstawowe etapy [Rijsbergen 1978]: generowanie reprezentacji dokumentu oraz indeksowanie. Tech-



niki te mogą również wykorzystywać metody zaczerpnięte z dziedziny sztucznej inteligencji (AI) [Maybury 1997].

Na etapie generowania reprezentacji dokumentu, zwanego również konflacją (ang. *conflation*), w sposób automatyczny tworzona jest taka postać dokumentu, która następnie może być wykorzystana przez system wyszukiwania informacji. Proces ten przebiega w następujących krokach: usunięcie słów o dużej częstości występowania, obcięcie sufiksów (przrostków) i wykrycie ekwiwalentnych stemów (tematów wyrazu).

Pierwszy krok polega na porównaniu tekstu z pewną listą słów o dużej częstości występowania zwaną 'stop-listą' i usunięciu tych słów z tekstu. Usunięcia dokonuje się ze względu na fakt, iż słowa te są słabymi dyskryminantami, przez co są mało przydatne w procesie wyszukiwania.

Drugi krok polega na usunięciu najdłuższego możliwego sufiksu, czego dokonuje się na podstawie tzw. tabeli sufiksów. Tabela ta jest zwykle znacznie dłuższa niż *stop lista* (ang. *stop list*) [Rijsbergen 1978]. W efekcie realizacji procesu usuwania sufiksów wyodrębniane zostają rdzenie, których długość musi przekraczać dwie litery, a ponadto muszą spełniać pewne dodatkowe warunki wynikające ze specyfiki języka (dla języka angielskiego np. taki, że ostatnią literą rdzenia nie może być litera „q”).

Trzeci i ostatni krok w procesie konflacji polega na wykryciu rdzeni ekwiwalentnych, przyjęto bowiem założenie, że jeśli dwa słowa mają ten sam rdzeń, to odpowiadają one jednemu, temu samemu, pojęciu. Założenie takie może być czasem zbyt wielkim uproszczeniem.

Cały proces generowania reprezentacji dokumentu może być obciążony błędami polegającymi na usunięciu niewłaściwego sufiksu bądź błędnego przypisania dwóch różnych słów do tego samego rdzenia. Jednak nawet metody stosowane 30 lat temu charakteryzowały się dopuszczalnymi poziomami błędów. Aby zredukować do minimum występowanie błędów, należy zastosować bardziej złożone metody, analizujące kontekst użycia danego słowa wraz z prawie kompletną listą słów oraz odpowiadających im rdzeni, co w przypadku współczesnych systemów oraz sieciowych zasobów, takich jak WordNet [Fellbaum 2001] nie nastręcza wielkich trudności. Oczywiście wykorzystane w procesie generowania reprezentacji dokumentu metody w dużej mierze zależą od języka, w którym jest on napisany, a także od dziedziny, której on dotyczy. Dla przykładu, w przypadku języków fleksyjnych, a takim jest m.in. język polski, drugi i trzeci krok powinny być raczej zastąpione bardziej złożoną analizą morfologiczną tekstu [Sobecki 1989].

Efektem realizacji omawianego etapu generowania reprezentacji dokumentu [Rijsbergen 1978] jest zbiór słów kluczowych lub termów, gdzie każde słowo kluczowe odpowiada jednemu i tylko jednemu elementowi ze zbioru zidentyfikowanych w dokumencie rdzeni.

Drugim etapem strukturalizacji jest indeksowanie. Indeksowanie (automatyczne) [Dąbrowski 1978] definiowane jest jako proces mający na celu zapis informacji zawartej w tekście danego dokumentu z użyciem pewnego języka informacyjnego. W wyszukiwaniu informacji język indeksowania służy także do wyrażania kwerend użytkownika. Języki indeksowania można podzielić na prekoordynowane (ang. *pre-coordinated*) i postkoordynowane (ang. *post-coordinated*). Pierwsze wskazują, że terminy są koor-

dynowane w czasie indeksowania, tzn. dowolna logiczna kombinacja terminów indeksowych może identyfikować pewną klasę dokumentów, natomiast drugie koordynowane są na etapie wyszukiwania przez łączenie klas dokumentów, które są zaetykietowane pojedynczymi terminami indeksowymi. Języki indeksowania mogą również wykorzystywać słownictwo kontrolowane i niekontrolowane. Pomimo wielu kontrowersji dotyczących najbardziej efektywnego języka indeksowania, już eksperymenty projektu SMART [Salton 1975] pokazały, że najprostsze procedury, które przyporządkowują dokumentowi lub kwerendzie pewien zbiór terminów, z wagami lub bez, pochodzących z treści dokumentu lub kwerendy są najbardziej efektywne. Stąd opracowane wiele lat temu proste metody indeksowania są nadal niezmiennie wykorzystywane we współczesnych systemach rekomendacyjnych.

Do najczęściej wykorzystywanych metod indeksowania stosowanych w systemach rekomendacyjnych należą: indeksowanie boolowskie i TF-IDF. Indeksowanie boolowskie jest bardzo prostą metodą indeksowania polegającą na tworzeniu reprezentacji dokumentu lub innych elementów za pomocą pewnego zbioru terminów (słów kluczowych), które go opisują.

Metoda indeksowania TFIDF (ang. *Term Frequency Inverted Document Frequency*) [Salton 1983] została opracowana na gruncie dziedziny IR. Metoda ta wykorzystuje ważony wektor cech, przedstawiony w punkcie 3.3.2.2, do reprezentacji profilu. Waga  $TFIDF_{w,S}$  słowa  $w$  w zbiorze  $S$  wyraża się następującym wzorem:

$$TFIDF_{w,S} = \frac{V'(w, S)}{\sqrt{\sum_{w' \in S} (V'(w', S))^2}}, \quad (1)$$

gdzie:  $V'(w, S) = \log(TF_{w,S} + 1) \log(IDF_w)$ ,

$TF_{w,S}$  – częstość występowania słowa  $w$  w zbiorze  $S$ ,

$N$  – liczność całej rozważanej kolekcji (np. dokumentów),

$IDF_w$  – odwrotność częstości występowania słowa  $w$  w całej kolekcji.

### 3.3.4.2. Grupowanie

Kolejną techniką uczenia profilu jest grupowanie (ang. *clustering*) i odnosi się głównie, choć nie wyłącznie, do grupowania użytkowników. Problem grupowania można zdefiniować jako taki podział danego zbioru na podzbiory, który optymalizuje określone kryterium. Dla przykładu, kryterium zniekształceń (ang. *distortion criterion*) może być określone jako pierwiastek ze średnich odległości euklidesowych (patrz 4.2.2) między grupowanymi elementami zbiorów reprezentowanymi np. przez ważne wektory cech, a odpowiadającymi im centrom grup, wykorzystywane w określeniu problemu  $k$ -średnich.

Po raz pierwszy problem podziału został sformułowany w pracy Steinhausa w 1956 r., gdzie postawione zostało następujące pytanie: „Jak należy wybrać  $n$  punktów z rozkładu dużej liczby punktów na płaszczyźnie, tak aby w najlepszy sposób reprezentowały wszystkie te punkty” [Steinhaus 1956]. Problem ten zwany jest obecnie „euklidesowym  $k$ -medianowym problemem” [Hochbaum 1982].

Obecnie stosowane metody grupowania zostały opracowane i są wykorzystywane głównie w ramach dziedzin IR oraz eksploracji danych (ang. *Data Mining*), maszynowego uczenia (ang. *Machine Learning*) i rozpoznawania wzorców (ang. *Pattern Recognition*). W ramach pierwszej z dziedzin częściowej, szczególnie w przeszłości, zamiast terminu *grupowanie* używano terminu *klasyfikacja* [Dąbrowski 1978], natomiast w ramach drugiej dziedziny stosuje się raczej termin *grupowanie* lub *analiza skupień*.

Wyróżnia się dwa rodzaje algorytmów grupowania: hierarchiczne i oparte na podziałach zbiorów [Daniłowicz 2002]. Algorytmy pierwszego rodzaju polegają na wyznaczeniu kolejnych grup (klastrow) na podstawie grup wcześniej utworzonych, a dokładniej tworzą one pewnego rodzaju hierarchię grup, bądź też podziałów, które można przedstawić jako drzewo [Wilson 2000], którego korzeń zawiera cały zbiór, który podlega podziałowi i każdy wierzchołek niższego poziomu (syn) jest podzbiorem wierzchołka wyższego poziomu (ojca), wierzchołki będące synami jednego ojca są zbiorami rozłącznymi oraz suma zbiorów wierzchołków liści równa jest całemu zbiorowi z korzenia. Algorytmy oparte na podziałach wyznaczają go „od razu”, tzn. bez wyznaczania podziałów pośrednich. Jednym z najbardziej znanych algorytmów jest algorytm rozwiązujący problem  $k$ -średnich (ang. *k-means*) znany również jako algorytm Lloyda lub prościej algorytmem  $k$ -means [Lloyd 1982], który z kosmetycznymi modyfikacjami w dziedzinie wyszukiwania informacji (IR) znany był jako algorytm Dattoli [Dattola 1968], [Dąbrowski 1978].

Problem  $k$ -średnich jest NP-trudny, ale istnieją rozwiązania, które pozwalają na znalezienie suboptymalnego rozwiązania o złożoności wielomianowej, np. algorytm Lloyda, który działa w następujących krokach [Kanago 2002]:

**Krok 1.** Losowo wybierz  $k$ -elementów jako centra grup, tzw. centroidy.

**Krok 2.** Przypisz każdy element zbioru do takiej grupy, do której element ten ma minimalną odległość liczoną do centroidu reprezentującego tę grupę.

**Krok 3.** Oblicz na nowo centroid każdej grupy na podstawie wszystkich elementów, które zostały do niej przypisane.

**Krok 4.** Skocz do kroku 2, o ile nie został spełniony określony warunek konwergencji, np. żaden z centroidów nie uległ zmianie od ostatniej iteracji.

Atrakcyjność algorytmu polega na jego prostocie i zdolności do zakończenia pracy w przypadku wykorzystania wymienionego warunku konwergencji oraz w przypadku, gdy nie występują elementy o równej odległości do dwóch lub więcej centroidów. Niestety, algorytm ten ma również pewne wady. Pierwsza wynika z jego suboptymalności i sprowadza się do wrażliwości na zakończenie działania w optimaach lokalnych zamiast globalnego. Druga wada, to pomimo wielomianowej złożoności obliczeniowej, długi czas jego działania z jednoczesnym brakiem możliwości skalowania, co może stanowić spory problem w przypadku dużych baz danych. Koszt Kroku 2 w każdej iteracji wynosi  $O(kdN)$ , gdzie  $k$  jest liczbą grup,  $d$  jest rozmiarem każdego elementu (np. wektora ważonego, który go reprezentuje), a  $N$  jest liczbą elementów, zakończenie zaś działania algorytmu wymaga zwykle wielu iteracji. Opracowano jednak wiele modyfikacji algorytmu, które działają o wiele szybciej, np. algorytm dwupodziału  $k$ -średnich (ang. *bisecting k-means*) [Kanago 2002], który rozpoczyna pracę od pojedynczego klastra za-

wierającego wszystkie elementy, następnie dzieli go na dwa klastry i zastępuje wejściowy podział przez podzielone klastry. Podział klastra polega na zastosowaniu algorytmu  $k$ -średnich  $\alpha$ -razy z wartością  $k = 2$ , zachowując ten podział, dla którego średnia odległość między wszystkimi elementami i centroidem jest najmniejsza.

Nieco innym, starszym i jednak mniej popularnym algorytmem uzyskania podziału zbioru elementów  $U = \{u_1, \dots, u_N\}$  reprezentowanych przez wektory ważne jest algorytm Dattoli [Dattola 1968, Dąbrowski 1978], który powstał w ramach dziedziny Wyszukiwania Informacji (IR). Algorytm ten jest bardzo podobny do algorytmu  $k$ -średnich. Kroki algorytmu są następujące [Dąbrowski 1978] (zachowano tu nazewnictwo określonych terminów):

**Krok 1.** Przyjmij a priori podział zbioru obiektów na  $k$ -klas początkowych  $C_{1,1}, C_{1,2}, \dots, C_{1,k}$ .

**Krok 2.** Wyznacz dla każdej iteracji  $t-1$  środki klas  $C_{t-1,1}, C_{t-1,2}, \dots, C_{t-1,k}$  o profilach  $O_{t-1,1}, O_{t-1,2}, \dots, O_{t-1,k}$ , gdzie każdy z profili  $O_{t-1,j} = (o_{j,1}, o_{j,2}, \dots, o_{j,d})$  dla  $j = 1, \dots, k$  jest wyznaczany na podstawie następującego wzoru:

$$o_{j,s} = \begin{cases} 0, & \text{jeżeli } f_{j,s} = 0 \\ b - r_{j,s} & \text{w innym przypadku} \end{cases} \quad (2)$$

gdzie:  $s = 1, \dots, d$ ,  $d$  – rozmiar każdego elementu,

$b$  – pewna stała przyjęta a priori oraz:

$$f_{j,s} = \sum_{u_i \in C_{t-1,j}} u_{i,s} \quad (3)$$

$$r_{j,s} = 1 + \max \{f_{j,t} : t = 1, 2, \dots, d\} - f_{j,s} \quad (4)$$

**Krok 3.** Utwórz nowe klasy  $C_{t,1}, C_{t,2}, \dots, C_{t,k}$  które wyznaczone są w następujący sposób:

$$C_{t,j} = \{u_i : \sigma(u_i, O_{t-1,j}) \geq T \text{ i } \sigma(u_i, O_{t-1,j}) = \max \{\sigma(u_i, O_{t-1,l}) \text{ dla } l = 1, 2, \dots, k\}\} \quad (5)$$

gdzie  $T$  jest pewnym progiem wykorzystywanym do tworzenia nowych klas. Wszystkie elementy, które nie zostały przyporządkowane do żadnej z klas zostają przyporządkowane do zbioru obiektów izolowanych  $L_t$ .

**Krok 4.** Oblicz nowe centroidy (tak jak w kroku 2)  $O_{t,1}, O_{t,2}, \dots, O_{t,k}$  sprawdzając następujący warunek:

$$\sum_{u_i \in C_{t,j}} \sigma(u_i, O_{t,j}) > \sum_{u_i \in C_{t,j}} \sigma(u_i, O_{t-1,j}), \quad (6)$$

i w przypadku gdy nie jest spełniony, wtedy  $O_{t,j} := O_{t-1,j}$ .

**Krok 5.** Jeżeli dla pewnego  $t$  i dla wszystkich  $j = 1, 2, \dots, k$  zachodzi  $O_{t,j} = O_{t-1,j}$ , to algorytm kończy iterację, a obiekty z  $L_t$  traktuje się jako izolowane lub przydziela się je

do klas, do których są najbardziej podobne, w przeciwnym przypadku  $t = t + 1$  przejdź do kroku 3.

W przypadku algorytmu Dattoli jako funkcje podobieństwa wykorzystywano funkcję P1 (patrz 4.2.2) dla  $c_1 = c_2 = n$ . Przydatność grupowania użytkowników uwidacznia się zwłaszcza wtedy, gdy istnieje spore zapotrzebowanie na dostarczenie rekomendacji w czasie rzeczywistym, gdyż o wiele szybsze jest porównanie profilu konkretnego użytkownika z pewną ograniczoną liczbą centroidów i wybór tej grupy, do której użytkownik pasuje najlepiej, niż znajdowanie dla tego użytkownika najbliższego sąsiedztwa, co wymaga wyznaczania podobieństwa ze wszystkimi użytkownikami zapisanymi w bazie. Takie podejście może również przyspieszyć dostarczenie samej rekomendacji, gdyż dla wyznaczonych wcześniej grup mogą zostać przygotowane off-line konkretne rekomendacje [Nguyen 2003a], [Sobecki 2004b].

### 3.3.4.3. Klasyfikatory

Klasyfikacją nazywamy przydział określonych obiektów do określonych uprzednio klas. Klasyfikacja, historycznie rzecz biorąc, była stosowana głównie w tzw. klasyfikacji biblioteczej. W późniejszym czasie znalazła zastosowanie w wyszukiwaniu informacji (IR) [Rijsbergen 1978], gdzie określana jest jako proces, w którym konstruuje się system klasyfikacyjny lub rezultat tego procesu (przy czym autor wyklucza wykorzystywane czasem w literaturze przedmiotu utożsamianie klasyfikacji z indeksowaniem dokumentów). W dziedzinie IR stosowano klasyfikację w grupowaniu (ang. *clustering*) słów kluczowych i grupowaniu dokumentów [Salton 1975]. Podobnie jak grupowanie, problem klasyfikacji był podejmowany przynajmniej od wczesnych lat 60. ubiegłego stulecia przez specjalistów wielu dziedzin, m.in. IR [Salton 1975]. Obecnie badania te są kontynuowane przez w specjalistów wielu dziedzin, w tym w ramach HCI, modelowania użytkowników i marketingu [Maglio 2000].

W ramach dziedziny IR przejrzysty i intuicyjny podział metod klasyfikacji przedstawiła Sparck-Jones [Rijsbergen 1978], wyróżniając następujące kryteria podziału uzupełnioną o jedną (ostatnią) kategorię:

- relacja między własnościami klasyfikowanych obiektów a klasami: jednokryterialne i wielokryterialne, gdzie klasa definiowana jest w odniesieniu do zbioru własności, które są zarówno konieczne, jak i wystarczające do określenia członkostwa obiektu w danej klasie.
- relacja między obiektami i klasami: wyłączne i nakładające się;
- relacja między samymi klasami: uporządkowane (np. hierarchicznie) i nieuporządkowane;
- sposób ich przeprowadzenia: manualna i automatyczna.

Przykładowo, UKD (Uniwersalna Klasyfikacja Dziesiętna), której początki sięgają prac Deweya z połowy XIX w. [Daniłowicz 1992], a której pierwsza wersja została opracowana przez Międzynarodowy Instytut Bibliograficzny w Brukseli w 1905 r. jest przykładem klasyfikacji nomotetycznej, wyłącznej, uporządkowanej – hierarchicznej i najczęściej realizowanej w sposób manualny.

W zastosowaniach systemów rekomendacyjnych do znajdowania rekomendowanych elementów na podstawie profilu użytkownika wykorzystuje się wiele różnych metod, w których profil stosowany jest jako klasyfikator, np.: sieci neuronowe, sieci bayesowskie, reguły asocjacyjne i drzewa decyzyjne (np. ID3 [Quinlan 1986]). Dla przykładu, w systemie rekomendacji filmów, który został przedstawiony w pracy [Sobecki 2006] zastosowano prostą sieć neuronową do klasyfikacji użytkowników na podstawie danych demograficznych obejmujących płeć, wiek i zawód użytkownika, do klas zainteresowań określonymi gatunkami filmów.

### 3.3.5. Informacja zwrotna od użytkownika

Sprzężenie zwrotne oparte na relewancji (ang. *relevance feedback*), tak jak i wiele innych, wcześniej omawianych metod, jest od wielu lat rozwijane na gruncie dziedziny IR [Salton 1975]. Tworzony najczęściej w momencie rejestracji lub pierwszego logowania model użytkownika już po pewnym czasie użytkowania staje się nieaktualny i wymaga aktualizacji. Sposobem zapewnienia dopływu informacji o bieżących preferencjach użytkownika jest sprzężenie zwrotne, realizowane najczęściej w drodze dostarczania przez użytkownika ocen prezentowanych elementów (problem ten został przedstawiony szerzej przy okazji omawiania danych użytkownika w punkcie 3.3.1.2).

Można wyróżnić cztery rodzaje sprzężenia zwrotnego [Montaner 2003]: brak informacji zwrotnej, sprzężenie explicite, sprzężenie implicate oraz sprzężenie hybrydowe.

W przypadku gdy nie występuje informacja zwrotna (**brak informacji zwrotnej**), wówczas żaden sposób nie są pozyskiwane informacje odnoszące się do oceny rekomendowanych elementów. Dopuszczana jest jednak manualna możliwość zmiany profilu i uzyskanie w ten sposób wpływu na modyfikację rekomendacji.

**Sprzężenie explicite** występuje wówczas, gdy użytkownicy są zobligowani do oceniania rekomendowanych elementów. Jest to metoda sprzężenia zwrotnego stosowana dość często. Zastosowano ją także w systemie informacyjnym *Megane* z rekomendacją interfejsu [Sobecki 2004b], w którym użytkownicy proszeni są o ocenę satysfakcji z interfejsu zastosowanego w tym systemie. W systemie *The Movies* [Sobecki 2006], na etapie tworzenia profilu, użytkownik ocenia standardowy zestaw filmów, a następnie może ocenić rekomendowane filmy.

Podstawową zaletą zastosowania tego typu sprzężenia zwrotnego jest jego prostota, dlatego jest ono wykorzystywane w wielu systemach rekomendacji, także w bardzo popularnym kolaboratywnym systemie rekomendacji filmów *MovieLens*. Potwierdzenie efektywności tego rozwiązania można już znaleźć w systemach wyszukiwania informacji implementowanych wiele lat temu [Salton 1975].

Sprzężenie zwrotne explicite ma jednak pewne wady, a mianowicie:

- nadawane oceny są subiektywne i relatywne, tzn. zależą od innych ocenianych równocześnie elementów;
- niektórym użytkownikom oceny wyrażone liczbowo mogą wydawać się nieadekwatne;



- często przyznawanym przez użytkowników ocenom nie można zbyt ufać [Kobsa 2001], ze szczególnym uwzględnieniem ocen negatywnych [Schwab 2000];
- użytkownicy niezbyt chętnie dostarczają jakichkolwiek ocen [Manning 2008], nawet wtedy, gdy są do tego zachęcani, w pracy [Panzzani 1996] szacuje, że jedynie 15% użytkowników dostarcza ocen, gdyż nie widzą bezpośredniego zysku z nakładu pracy na ich dostarczenie.

**Sprzężenie implicite** występuje wówczas, gdy oceny rekomendowanych elementów nie są dostarczane bezpośrednio przez użytkownika, ale system sam wywodzi te oceny z różnych jego działań. Podobnie jak sprzężenie explicite, pojęcie sprzężenia implicite zostało zdefiniowane wiele lat temu [Rich 1979]. Wykorzystując tę metodę można pokonać pewne niedogodności zastosowania sprzężenia explicite, takie jak niewielka liczba ocen, czy relatywność nadawanych ocen. Różne metody zbierania informacji implicite, takie jak akcje selekcji, czas oglądania strony, realizowane zakupy itp. zostały przedstawione w punkcie 3.3.1.2.

Ostatnim z wymienionych w pracy [Montaner 2003] typem sprzężenia zwrotnego jest **sprzężenie hybrydowe**, w którym mechanizm sprzężenia explicite jest uzupełniany o dodatkowe akcje implicite. Dla przykładu, w systemie rekomendacji filmów w wypożyczalni wideo, jakim jest np. system *The Movie* [Nguyen 2007], możliwość dokonania oceny rekomendowanego filmu bezpośrednio przez użytkownika (explicite) można uzupełnić o parametr określający taką ocenę jako wiarygodną na podstawie stwierdzenia, iż film taki został wypożyczony, a sama ocena nastąpiła po jego obejrzeniu, o czym świadczyłaby data zwrotu filmu (akcja implicite).

### 3.3.6. Techniki adaptacji profilu

Systemy rekomendacyjne mogą być używane przez poszczególnych użytkowników w dłuższym czasie, dlatego konieczne jest zastosowanie pewnej techniki, która umożliwiałaby dostosowanie ich do zmieniających się potrzeb użytkownika. Zakłada się, że nowo zebrane dane użytkownika, przede wszystkim w drodze sprzężenia zwrotnego opisanego w punkcie 3.3.5, w lepszy sposób reprezentują preferencje użytkownika niż dane zebrane wcześniej. Na podstawie założenia opracowano wiele technik umożliwiających dokonanie modyfikacji profilu użytkownika [Montaner 2003]. Można wyróżnić następujące typy techniki adaptacji: manualna, adaptacji po dodaniu nowej informacji, funkcji stopniowego zapominania i naturalna selekcja.

Najprostszym sposobem adaptacji profilu jest zastosowanie **manualnej** jego aktualizacji. Jest ona wykorzystywana przede wszystkim w systemach, w których profil jest inicjowany ręcznie. Wtedy zwykle umożliwia się także jego modyfikację na dalszych etapach pracy z systemem. W systemie np. *Kucharek* [Sobecki 2006a] użytkownik określający podczas tworzenia swojego profilu jaki rodzaj kuchni preferuje (jarską lub mięsną), może wybór ten w przyszłości zmodyfikować manualnie, co będzie oczywiście odzwierciedlone w profilu i będzie miało wpływ (w tym przypadku bardzo istotny) na rekomendowane potrawy. Mimo swojej prostoty ten typ adaptacji ma wszelkie wady

manualnych czynności przytoczone w punktach 3.3.1.2, 3.3.3 i 3.3.5, jest jednak wykorzystywany (oczywiście niewyłącznie) w większości systemów rekomendacyjnych

Inną bardzo często wykorzystywaną we współczesnych systemach rekomendacyjnych techniką adaptacji jest tzw. **adaptacja następująca po dodaniu nowej informacji**. Dla przykładu, w systemach rekomendacji filmów, np. *The Movie* [Nguyen 2007], dostarczenie każdej oceny filmu zmienia profil tego użytkownika, a przez to zmienia zarówno (czasem jedynie nieznacznie) grupę podobnych użytkowników (komponent kolaboracyjny), jak i preferencje użytkownika dotyczące gatunków filmów (komponent bazujący na zawartości). Oba komponenty zostaną szczegółowiej omówione w punkcie 4.1. Wadą tego rozwiązania jest brak możliwości zapominania starszych wpisów [Montaner 2003].

Kolejną techniką adaptacji stosowaną w systemach rekomendacyjnych jest **funkcja stopniowego zapominania**. Idea wykorzystania stopniowego zapominania pojawiła się w połowie lat dziewięćdziesiątych ubiegłego stulecia. W pracach [Maloof 1995], [Montaner 2003] proces zapominania określa się jako proces stopniowy. W pracy [Koychev 2000] przedstawiona została liniowa funkcja stopniowego zapominania, która modyfikuje wartości pewnych akcji na podstawie wag uwzględniających upływ czasu od nadania tych wartości. Szczególnym przypadkiem stopniowej funkcji zapominania jest podejście wykorzystujące okno czasu (ang. *time window*), które bazuje na tworzeniu modelu jedynie na podstawie obserwacji pochodzących z określonego przedziału czasu (np. wyłącznie najnowsze obserwacje), z całkowitym pominięciem wszystkich obserwacji spoza wybranego przedziału czasu [Maloof 2000].

Ostatnim typem adaptacji profilu jest tzw. **naturalna selekcja** i jest ona związana z wykorzystaniem technologii multiagentowych. Metafora naturalnej selekcji zakłada, że agenty w systemie multiagentowym, tworzącą pewnego rodzaju ekosystem [Montaner 2003]. Ekosystem ten może ewoluować w ten sposób, że najlepsze agenty (czyli takie, które dostarczają najlepsze rekomendacje) pozostają w ekosystemie, natomiast mniej efektywne są niszczone, co jest nieco innym podejściem do ekosystemu agentów, niż metody consensusu prezentowane m.in. w pracach [Nguyen 2003a] i [Nguyen 2008], gdzie pod uwagę brane są wszystkie opinie agentów.

### 3.4. Podsumowanie

W rozdziale przedstawiono problemy modelowania użytkowników systemów rekomendacyjnych. W ramach rozdziału przedstawiono dziedziny zastosowań systemów rekomendacyjnych, cele i zadania użytkowników w systemie rekomendacyjnym, oraz zawartość, reprezentacja, tworzenie i aktualizacja profilu użytkownika.

Dziedziny zastosowań systemów rekomendacyjnych w dużej mierze pokrywają się z dziedzinami zastosowań systemów webowych. Systemy rekomendacyjne można zastosować wszędzie tam, gdzie można poprawić efektywność pracy systemu przez ułatwienie użytkownikowi dokonania decyzji wyboru. Także prace Autora dotyczące systemów rekomendacyjnych obejmują różnorodne dziedziny zastosowań, są to rekomen-

dacja filmów [Sobecki 2006], [Rakowski 2006] i [Nguyen 2007], rekomendacja przepisów kulinarnych [Sobecki 2006] i [Sobecki 2006a], rekomendacji interfejsów użytkownika [Sobecki 2003a], [Sobecki 2004], [Sobecki 2004a], [Sobecki 2004b], [Sobecki 2005], [Sobecki 2007], [Sobecki 2007a] i [Sobecki 2008a], rekomendacji wiadomości [Sobecki 2007] i prezentów [Sobecki 2009], przy czym pewne prace dotyczyły zarówno rekomendacji elementów, jak i interfejsu użytkownika.

W celu zamodelowania zachowań użytkownika w systemie interakcyjnym należy poznać cele, jakie ma zamiar realizować użytkownik za pomocą tego systemu. Oczywiście podstawowym celem użytkowników każdego systemu rekomendacyjnego jest uzyskanie rekomendacji związanej z dziedziną zastosowania określonego systemu. Cel ten realizowany jest przez systemy rekomendacyjne dotyczące wszystkich zastosowań, oprócz rekomendacji interfejsu użytkownika, gdyż dla większości użytkowników tych systemów rekomendacja ta dokonuje się bez jej uświadomienia przez większość użytkowników. Użytkownicy tych systemów mogą również chcieć zweryfikować jakość dostarczanych rekomendacji. Kolejny cel użytkowników, a mianowicie tuningowanie profilu może dotyczyć użytkowników wszystkich systemów. Trzy ostatnie, wyrażenie siebie, pomoc innym i wywarcie wpływu na innych dotyczy wszystkich dziedzin zastosowania systemów, jednak takich, które wykorzystują m.in. filtrowanie kolaboratywne, czyli większość z zaprezentowanych systemów.

W kolejnym punkcie niniejszego rozdziału przedstawiono problemy modelowania użytkowników systemów rekomendacyjnych, a zwłaszcza zawartość profilu, jego reprezentacja, tworzenie i aktualizacja. Problemy te oprócz metod wykorzystania profilu użytkownika stanowią istotę systemów rekomendacyjnych. W niniejszym rozdziale oprócz stanu wiedzy w zakresie modelowania użytkowników przedstawiono wyniki prac Autora, a zwłaszcza metodę reprezentacji profilu w postaci krotki [Sobecki 2001] i [Nguyen 2003a], struktury drzewiastej [Sobecki 2002b], w postaci dokumentu XML [Sobecki 2003].

Oprócz metody reprezentacji profilu użytkownika, bardzo istotnym elementem modelowania użytkowników są sposoby inicjowania i uczenia profilu. Profil początkowy może być pusty lub inicjowany na trzy sposoby: manualnie, na podstawie stereotypów lub zbioru uczącego. W pracach autora zastosowano wszystkie cztery metody. Dla przykładu w systemie *Megane* [Sobecki 2004b], zastosowano m.in. manualny sposób inicjowania profilu, gdyż zawierał on dane demograficzne, w systemie *Kucharek* [Sobecki 2006a] na podstawie stereotypów z danych demograficznych określone są wartości profilu użytkownika służące do określenia interfejsu użytkownika, natomiast w jednym z systemów zaprezentowanych w pracy [Sobecki 2006], wykorzystano zestaw uczący utworów muzycznych do określenia preferencji muzycznych użytkownika. Bardzo często zastosowana metoda inicjowania profilu zależy od specyfiki dziedziny implementacji oraz przewidywanego przez projektanta stopnia, w którym użytkownicy zechcą uczestniczyć w procesie tworzenia swojego profilu.

Kolejnym elementem tworzenia profilu jest jego uczenie. W niniejszym rozdziale przedstawiono trzy podstawowe metody: strukturalizację, grupowanie i klasyfikację.

Stosowane w RS metody strukturalizacji informacji wypracowane zostały na gruncie IR. Prace autora w tym zakresie koncentrowały się przede wszystkim na wykorzystaniu systemu WAIS do implementacji wyszukiwania pełnotekstowego [Jakubowski 1996], [Jakubowski 1997] i [Sobecki 1997]. Grupowanie i klasyfikację wykorzystywano np. w systemie *Megane* [Sobecki 2004b], który bardziej szczegółowo opisano w rozdziale 5.

Ostatnimi rozważanymi zagadnieniami związanymi z tworzeniem i utrzymaniem profilu użytkownika jest adaptacja profilu i zastosowanie zwrotnej informacji o relewancji rekomendowanych elementów. Adaptacja profilu może być realizowana na cztery sposoby. Najprostszym jest modyfikacja manualna, wykorzystywana w większości systemów opisywanych w pracach autora, np. wspomniane systemy *Megane* i *Kucharek*. Również druga metoda, a mianowicie po dodaniu nowej informacji realizowana jest również w systemie *The Movie* [Nguyen 2007], w którym po dodaniu pewnej liczby ocen obliczane na nowo jest sąsiedztwo określonego użytkownika, a co za tym idzie przeliczana jest predykcja ocen rekomendowanych filmów. Dwie inne metody adaptacji to naturalna selekcja i funkcja stopniowego zapominania, które np. zastosowano w systemie *The Movie* [Nguyen 2007]. Sprzężenie zwrotne relewancji rekomendowanych elementów może być realizowane explicite, implicite oraz hybrydowo. W pracach Autora wykorzystywane jest przede wszystkim sprzężenie explicite, w którym użytkownik określa manualnie stopień relewancji rekomendowanego elementu, a czasem nawet wypełnia bardziej złożoną ankietę określającą np. satysfakcję z zaproponowanego interfejsu użytkownika, np. w systemie *Megane* [Sobecki 2004b].



## 4. Wykorzystanie modelu użytkownika w systemach rekomendacyjnych

Faza wykorzystania profilu w systemie rekomendacyjnym następuje po wcześniejszym utworzeniu profilu wraz z udostępnieniem danych i struktur dotyczących użytkowników i rekomendowalnych elementów, co umożliwia dalsze wnioskowanie w określonej dziedzinie zastosowania. Kategoria wykorzystania profilu użytkownika w taksonomii systemów rekomendacyjnych [Montaner, 2003] została podzielona na następujące podkategorie: metoda filtrowania informacji, technika przyporządkowania rekomendowanego elementu do profilu użytkownika oraz technika wyznaczania podobieństwa między użytkownikami. Ogólny schemat wykorzystania profilu użytkownika do rekomendacji przedstawia rys. 10, będący modyfikacją rysunku zamieszczonego w pracy [Montaner 2009].

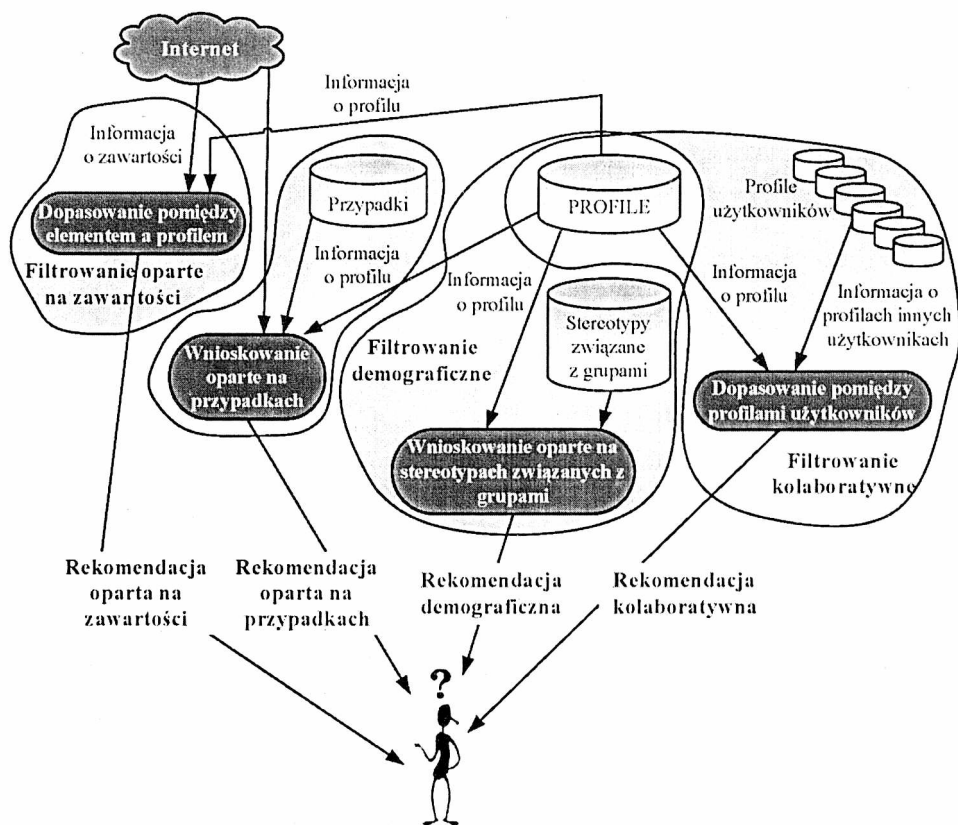
### 4.1. Metoda filtrowania informacji

Metoda filtrowania informacji umożliwia wyselekcjonowanie (przefiltrowanie) tych spośród nich, które umożliwią podjęcie decyzji dotyczącej rekomendacji. W pracy [Montaner 2003] wyróżniono trzy podstawowe metody filtrowania: demograficzną (ang. *demographic filtering* – DF), kolaboratywną (ang. *collaborative filtering* – CF) oraz opartą na zawartości (ang. *content-based filtering* – CBF) i hybrydową (ang. *hybrid approach* – HA). Ponadto w literaturze przedmiotu można znaleźć również i inne metody filtrowania, takie jak: bazującą na przypadkach użycia (ang. *case-based reasoning* – CBR) lub ich modyfikacje konwersacyjne CBR (ang. *conversational CBR* – CCBR) [Aktas 2004] oraz modyfikacja CF biorąca za podstawę wnioskowania związku między elementami, a nie użytkownikami (ang. *item based CF* – IBCF) [Ahn 2007].

#### 4.1.1. Filtrowanie demograficzne (DF)

Podejście demograficzne charakteryzuje się wykorzystaniem wnioskowania stereotypowego oraz danych demograficznych zawartych w profilu użytkownika do znalezienia relacji między użytkownikiem a rekomendowanym obiektem. Przyporządkowań takich można dokonywać w sposób bezpośredni, np. kraj pochodzenia może wyznaczać język wykorzystywany do prezentacji informacji, dokładne zaś miejsce zamieszkania użytkownika może określać region, dla którego będzie prezentowana prognoza pogody.





Rys. 10. Wykorzystanie profilu użytkownika do rekomendacji

Wnioskowanie stereotypowe jest uznawane za problem klasyfikacji [Kobsa 2001]. Podstawowym celem wnioskowania stereotypowego wykorzystywanego w DF jest generowanie predykcji dotyczących użytkownika, tak jak przytoczone w punkcie 3.3.3.1 dane dotyczące kodu pocztowego pozwalają na przyjęcie pewnych założeń dotyczących statusu społecznego mieszkańców tego miejsca [Quinn 2001]. DF jest dość proste w implementacji i pozwala na dostarczanie w miarę efektywnej rekomendacji nawet w przypadku skąpych informacji dotyczących użytkownika. Podejście demograficzne ma jednak pewne istotne wady, a mianowicie:

- bazuje ono jedynie na generalizacji zainteresowań użytkowników, dostarczając tożsamych rekomendacji dla użytkowników z podobnymi demograficznymi danymi, mimo że każdy użytkownik jest nieco inny;
- uniemożliwia zastosowanie indywidualnych rekomendacji dla użytkowników;
- uniemożliwia adaptację rekomendacji w czasie.

DF zostało z powodzeniem zastosowane w wielu systemach rekomendacyjnych. W webowym systemie informacyjnym *Megane* [Sobecki 2004b] wnioskowanie stereotypowe pozwala ekspertom dokonać przyporządkowania użytkownikom identyfikowanym przez profil zawierający dane demograficzne odpowiednich parametrów interfejsu

użytkownika. W innym systemie, rekomendującym przepisy kulinarne *Kucharek* [Sobecki 2006a], DF wykorzystuje wnioskowanie rozmyte [Mamdani 1974] z późniejszymi modyfikacjami [Sugeno 1985]. Główną zaletą takiego rozwiązania jest dostarczenie ekspertom dziedzinowym możliwości bardziej precyzyjnego narzędzia modelowania preferencji użytkowników niż klasyczne wnioskowanie stereotypowe. W systemie *Kucharek* wybrane zostały określone atrybuty demograficzne opisujące użytkownika, takie jak: wiek, płeć, doświadczenie w gotowaniu oraz preferencje kulinarne, które pozwoliły na stworzenie 170 reguł wnioskowania rozmytego rekomendujących określone przepisy kulinarne.

#### 4.1.2. Filtrowanie bazujące na zawartości (CBF)

Drugim typem filtrowania jest CBF, które bazuje na wcześniej dokonanych ocenach (implicite lub explicite) określonych elementów. Ocenę tę, zapisaną są oczywiście w profilu użytkownika w tzw. części danych użytkownika. Przykładowo, dane użytkownika mogą zawierać URL odwiedzanych stron dających podstawę do wnioskowania, iż użytkownik zainteresowany jest treściami prezentowanymi na tych stronach. Przykładem zastosowania tego rozwiązania może być wspomniany już system *Letizia*, autonomiczny agent wspomagający przeglądanie sieci WWW [Lieberman 1997], który śledzi odwiedzane strony użytkownika, odnotowuje słowa kluczowe znajdujące się na odwiedzanych stronach, a następnie na nowe zapytania użytkownika kierowane do systemu w dalszej jego pracy rekomenduje kolejność relewantnych stron. Innym przykładem agenta tego typu jest *Apt Decision* [Shearin 2001], który wykorzystuje opisy lokalii, z którymi użytkownik już się zapoznał do sortowania nowych rekomendacji.

Pomimo wielu zalet rekomendacje bazujące na zawartości mają kilka wad [Montaner 2003]:

- CBF bazuje na obiektywnym opisie elementów rekomendowanych, natomiast sama ich selekcja przez użytkowników bardzo często oparta jest na subiektywnej ocenie ich zawartości,
- ma tendencję do zbyt dużego skupiania odpowiedzi, gdyż systemy tego typu nie mają wbudowanego sposobu odnajdywania elementów niespodziewanych (ang. *serendipitous*), mają więc tendencję do wynajdowania tylko takich elementów, które są podobne do już wcześniej ocenionych przez użytkownika,
- użytkownicy bardzo niechętnie podają swoje oceny, tak więc systemy często skazane są na automatyczne wyciąganie wniosków co do oceny użytkownika, a te bardzo często nie są zbyt dokładne.

#### 4.1.3. Filtrowanie kolaboratywne (CF)

Filtrowanie kolaboratywne charakteryzuje się przyporządkowaniem użytkowników do pewnych grup i rekomendowaniem takich elementów, które zyskały uznanie u innych użytkowników z danej grupy. Przynależność do określonych grup użytkowników określana jest na podstawie podobieństw między użytkownikami wyznaczanych np. na podstawie ocen nadawanych przez użytkowników tym samym elementom. Rekomen-

daje są zwykle wyznaczane na podstawie statystycznej analizy wzorców i analogii danych wydobytych ocen pewnych elementów dostarczanych przez użytkowników [Montaner 2003], przy czym oceny te mogą zostać uzyskane *explicite* lub *implicite*. Tak więc podstawową różnicą między podejściem bazującym na zawartości i podejściem kolaboratywnym jest szukanie podobieństw między rekomendowanymi elementami w pierwszym przypadku, a innymi użytkownikami w drugim.

Niektórzy autorzy wyróżniają oprócz standardowego CF również dwa inne typy filtrowania kolaboratywnego [O'Donovan 2004]: filtrowanie kolaboratywne bazujące na związkach między elementami (ang. *Item Based CF* – IBCF) oraz filtrowanie kolaboratywne bazujące na regułach asocjacyjnych (ang. *Association-Rule Based Filtering* – RBCF). Dla podkreślenia, że mamy do czynienia ze standardowym CF można posłużyć się nazwą: filtrowanie kolaboratywne bazujące na związkach między użytkownikami (*User Based CF* – UBCF) lub inaczej „czyste” filtrowanie kolaboratywne (ang. *Pure CF*).

Przykładem aplikacji opartej na podejściu kolaboratywnym mogą być *MovieLens* i *GroupLens* [Miller 1997]. Innym przykładem może być system *Doppelgänger* opisany w pracy [Kobsa 2001] lub szerzej opisany w dalszej części pracy system *Reporter* [Sobecki 2007], który do rekomendacji wiadomości wykorzystuje idee sztucznego systemu immunologicznego (AIS). Możliwe jest również grupowanie użytkowników na podstawie ich danych demograficznych lub szerzej danych użytkownika [Nguyen 2003], [Sobecki 2004b], a nie użytkowania, tak jak w podanych systemach lub np. w sklepie internetowym *Amazon.com*.

Aby systemy wykorzystujące podejście kolaboratywne dostarczały predykcji o zadowalającej dokładności (patrz punkt 4.4.2.1), muszą być spełnione następujące warunki: liczba użytkowników systemu musi być odpowiednio duża (aby zwiększyć prawdopodobieństwo, że każdy przychodzący użytkownik znajdzie grupę podobnych do siebie); należy znaleźć sposób na reprezentację zainteresowań użytkowników; należy dysponować algorytmami, które grupują użytkowników o podobnych zainteresowaniach.

Algorytm CF działa w następujących krokach [O'Donovan 2003]:

**Krok 1.** Obliczenie podobieństwa między „aktywnym” użytkownikiem i resztą użytkowników przez zastosowanie wybranej funkcji podobieństwa (patrz punkt 4.2.2).

**Krok 2.** Utworzenie grupy badawczej przez wybór pewnej liczby najbardziej podobnych użytkowników.

**Krok 3.** Wybór elementów ocenianych wysoko przez użytkowników z grupy badawczej, ale nieocenionych przez aktywnego użytkownika.

**Krok 4.** Zwrócenie uporządkowanej listy tych elementów jako zbioru rekomendowanego.

Podsumowując, CF ma wiele zalet, do których należą [O'Donovan 2003]:

- możliwość dostarczenia rekomendacji elementów pomimo braku obiektywnego ich opisu,
- umożliwienie rekomendacji tzw. „niespodziewanych” (ang. *serendipitous*), jednak relevantnych elementów,
- nieskomplikowane metody filtrowania.

Niestety, stosowanie algorytmu CF wiąże się także z pewnymi wadami, wśród których należy wymienić:

- problem użytkowników, którzy korzystając u zarania funkcjonowania systemu napotykać zupełny brak ocen lub też niedostateczną liczbę ocen nadanych przez innych użytkowników (ang. *sparsity problem*); sposobem radzenia sobie w takiej sytuacji jest dostarczenie rekomendacji przez ekspertów na podstawie zaklasyfikowania użytkownika do odpowiedniej klasy [Nguyen 2003], [Sobecki 2004b],
- problem użytkowników, których potrzeby lub gusty odbiegają w sposób znaczący od ludzi do niego podobnych, co najczęściej prowadzi do niezadowolających rekomendacji,
- osiągnięcie krytycznej masy uczestników biorących w eksperymencie jest bardzo kosztowne i ma szansę na powodzenie jedynie w przypadku bardzo popularnych systemów,
- krytyczna zależność między rozmiarem i złożonością populacji użytkowników wpływa na grupy najbliższych sąsiadów, tzn. użytkownik nawet po określeniu swojej niechęci wobec określonego elementu będzie dalej go otrzymywał, gdyż tak wskazuje grupa,
- ostatnią wadą tego rozwiązania podnoszoną przez specjalistów z dziedziny systemów rekomendacyjnych (RS) jest brak przejrzystości w dostarczanych rozwiązaniach; systemy tego typu mogą być postrzegane jako „czarne skrzynki” lub wyrocznie [Montaner 2003].

#### 4.1.4. Inne metody bazujące na filtrowaniu kolaboratywnym

Oprócz tzw. „czystego” filtrowania kolaboratywnego można wymienić: filtrowanie kolaboratywne bazujące na związkach między elementami (ang. *item based CF* – IBCF) oraz filtrowanie kolaboratywne bazujące na regułach asocjacyjnych (ang. *Association-Rule Based Filtering* – RBCF).

IBCF tym różni się od klasycznego algorytmu CF, że predykcje dokonywane są na podstawie podobieństw między elementami, a nie użytkownikami [Sarwar 2001]. Algorytm IBCF dla predykcji oceny określonego elementu przez danego użytkownika wybiera pewną liczbę ocenionych przez tego użytkownika elementów, a następnie wybiera te, które najbardziej są podobne do określonego elementu. Można więc wyróżnić dwie podstawowe fazy działania algorytmu IBCF: w pierwszej budowany jest model, a w drugiej wyznaczana jest predykcja [O'Donovan 2003].

Pierwsza faza budowy modelu polega na wyznaczeniu wartości podobieństwa między poszczególnymi elementami i umieszczeniu ich w kwadratowej macierzy podobieństw  $P$ , gdzie każdy element macierzy  $p_{ij}$  określa podobieństwo między elementem  $i$  oraz  $j$  dla  $i, j = 1, 2, \dots, n$ . Podobieństwo wyznacza się na podstawie ocen użytkowników, którzy ocenili obydwa produkty i np. współczynnika Pearsona [Pearsons 2006]. Tworzenie modelu odbywa się zwykle off-line.

Algorytm wyznaczania rekomendacji metodą IBCF realizowany jest w następujących krokach [O'Donovan 2003]:

**Krok 1.** Określony użytkownik dostarcza oceny zbioru elementów  $U$ .

**Krok 2.** Zbiór kandydatów  $C$  formowany jest przez zsumowanie  $k$  najbardziej podobnych elementów do każdego elementu  $j \in U$ , gdzie podobieństwo wyznaczone jest na podstawie macierzy podobieństw  $P$ .

**Krok 3.** Ze zbioru  $C$  należy usunąć wszelkie elementy należące również do zbioru  $U$ .

**Krok 4.** Znaleźć podobieństwo między każdym elementem  $c \in C$  a wszystkimi elementami ze zbioru  $U$ , które wyznaczone jest na podstawie macierzy podobieństw  $P$ .

**Krok 5.** Posortować zbiór  $C$  według nierosnących wartości podobieństw.

**Krok 6.** Pierwsze  $n$  elementów z posortowanego zbioru  $C$  jest zbiorem rekomendowanych elementów.

Drugą metodą wykorzystującą filtrowanie jest metoda RBCF, która jest podobna do IBCF, z tą różnicą, że wykorzystuje techniki eksploracji danych (ang. *data mining*), zwłaszcza algorytm wydobywania reguł asocjacyjnych Apriori występujących między elementami [O'Donovan 2004].

Reguły asocjacyjne [Agrawal 1993] są postaci  $X \Rightarrow Y$ , gdzie  $X$  i  $Y$  są kompleksami (zbiorami transakcji) złożonych z podzbioru zbioru elementów  $I = \{i_1, i_2, \dots, i_m\}$  przy czym dodatkowo zachodzi warunek  $X \cap Y = \emptyset$ . Reguły asocjacyjne mają poziomy wsparcia (ang. *support*) i wiarygodności (ang. *confidence*). Wsparcie reguły  $X \Rightarrow Y$  na zbiorze przykładów  $P$  zdefiniowane jest [Cichosz 2000] jako stosunek liczby przykładów ze zbioru  $P$  pokrywanych przez kompleksy  $X$  i  $Y$  do liczby wszystkich przypadków w tym zbiorze. Wiarygodność reguły  $X \Rightarrow Y$  na tym samym zbiorze przykładów jest określona jako stosunek liczby przykładów z tego zbioru pokrywanych przez kompleksy  $X$  i  $Y$  do liczby przykładów z tego zbioru pokrywanych przez kompleks  $X$ . Algorytm wyznaczania reguł asocjacyjnych Apriori [Agrawal 1994] opiera się na wykorzystaniu właściwości tzw. częstych kompleksów [Cichosz 2000], która mówi, że każdy kompleks zawierający się w częstym kompleksie jest również częstym kompleksem, dlatego zaczynając budowę od kompleksów jednoelementowych (atomowych) w każdym kroku dodaje się jeden dodatkowy element, gdyż jedynie nadzbiory częstych kompleksów są częstymi kompleksami.

#### 4.1.5. Wnioskowanie oparte na przypadkach użycia (CBR)

Oprócz podejść podstawowych, takich jak DF, CBF i CF istnieje również metoda rekomendacji CBR wykorzystująca wnioskowanie oparte na przypadkach (ang. *case-based reasoning*). Metoda CBR jest jedną z metod AI o szczególnej przydatności w rozwiązywaniu problemów, dla których brak jest odpowiednio ugruntowanych heurystyk, a istnieją jednak rozwiązane poprawnie i niepoprawnie przypadki [Grupe 1998]. Podstawą metody CBR jest baza przypadków, która przechowuje przeszłe przypadki [Oh 2007]. Metoda CBR znalazła zastosowanie w wielu różnych dziedzinach, m.in. w medycynie [Bichindaritz 2006], finansach [Oh 2007] i wytwarzaniu oprogramowania [Grupe 1998]. W przypadku systemów rekomendacyjnych CBR wykorzystuje informacje dotyczącą kontekstu użycia, np. platformy systemowej i sprzętowej, z której korzysta użytkownik podczas pracy z systemem rekomendacyjnym, miejsca wykorzystania lub najbliższego otoczenia użytkownika. Przykładami takich systemów mogą być systemy re-

komendujące określone elementy (np. restauracje) w zależności od aktualnego położenia użytkownika korzystającego z systemu na platformie mobilnej lub oferujące określone opcje systemu w zależności od określonej sytuacji (np. w samochodowym systemie nawigacyjnym inne opcje są udostępnione gdy samochód jest w ruchu, a inne gdy samochód się nie porusza). Kolejnym przykładem mogą być systemy adaptujące interfejs użytkownika w zależności od rodzaju i cech platformy (np. rozdzielczości ekranu).

#### 4.1.6. Podejście hybrydowe (HA)

Większość wad omówionych metod filtrowania można usunąć przez zastosowanie metody hybrydowej. Przykładowo, wadę filtrowania kolaboratywnego polegającą na niewielkiej liczbie zarejestrowanych użytkowników, co zwykle występuje na wczesnych etapach pracy systemu, można pokonać stosując filtrowanie demograficzne. Do rekomendacji interfejsu użytkownika w webowych systemach informacyjnych [Nguyen 2003] i [Sobecki 2004b] zastosowano metodę hybrydową, będącą połączeniem DF i CF, przy czym DF stosowano, gdy w systemie było zarejestrowanych niewielu użytkowników, a dopiero po przekroczeniu pewnej ich liczby zastosowano filtrowanie metodą CF. Inną metodą pokonania niedoskonałości podejścia CF, zwłaszcza problemu niewielkiej liczby ocen, może być zastosowanie rekomendacji bazującej na metodzie CBF, opartej na w miarę pełnym zestawie informacji dotyczących poszczególnego użytkownika [Montaner 2003]. Wśród zalet metody CBF należy wymienić fakt, iż potrafi ona dostarczyć zadowalających rekomendacji dla użytkowników o specyficznym guście, a również i to, że liczba użytkowników zarejestrowanych w systemie nie ma wpływu na jakość rekomendacji.

Wady podejścia bazującego na zawartości można z kolei pokonać stosując filtrowanie kolaboratywne CF. Do takich wad należy np. brak subiektywnej wiedzy na temat rekomendowanych elementów, którą to wiedzę z dużym powodzeniem dostarczają inni użytkownicy. W analogiczny sposób można pokonać inną wadę filtrowania metodą CBF, a mianowicie trudności z rekomendacją nowych elementów, dla których określony użytkownik nie posiada żadnych ocen.

Można wyróżnić przynajmniej dwa podstawowe typy podejścia hybrydowego [Sobecki 2006]: pierwsze tworzy rozwiązania na podstawie każdej z metod osobno oraz drugie, które integruje wiedzę pochodzącą z poszczególnych metod przed wyznaczeniem rekomendacji. Dla przykładu, stosując pierwsze podejście należy wyznaczyć rekomendację dla każdego z podejść DF, CBF i CF z osobna a następnie zintegrować listy odpowiedzi w pojedynczą listę. Następnie, korzystając z drugiego podejścia można modyfikować wnioskowanie stereotypowe lub wnioskowanie rozmyte będące podstawą metody DF, używając wiedzy pochodzącej od innych użytkowników (CF) lub wiedzy zgromadzonej podczas użycia danego systemu przez tego samego użytkownika (CBF).

W literaturze [Setten 2005] można spotkać również inne taksonomie metod hybrydowych, które obejmują metody: ważone, przełącznikowe, mieszane, łączone, kaskadowe, połączone i metapoziomowe.

Podejście hybrydowe jest obecnie podejściem dominującym spośród praktycznych realizacji systemów rekomendacyjnych. Wśród popularnych systemów rekomendacyj-



nych stosujących to podejście, można wymienić systemy *CDNow* oraz *Amazon.com* [Montaner 2003] lub różne systemy rekomendacyjne opisane w pracach [Sobecki 2005] i [Sobecki 2008b]. Bardziej szczegółowo różne metody hybrydowe omówiono w rozdziałach 5 i 6.

## 4.2. Techniki dopasowania rekomendowanego elementu do profilu użytkownika

Profil użytkownika w systemie rekomendacyjnym budowany jest w celu dostarczania rekomendacji. W przypadku niektórych metod filtrowania, takich jak DF, CBF i CBR wykorzystuje się bezpośrednio porównanie profilu i rekomendowanego elementu. W pracy [Montaner 2003] wyróżniono następujące techniki dopasowywania rekomendowanego elementu do profilu użytkownika: dokładne dopasowanie na podstawie słów kluczowych, podobieństwo cosinusowe [Salton 1975], najbliższy sąsiad (ang. *nearest neighbour*) i klasyfikatory. Zestaw ten jest tylko pewnym podzbiorem możliwych i bardzo często stosowanych sposobów znajdowania dopasowania między profilem użytkownika a elementem. Techniki dopasowywania zostały opracowane, często dziesiątki lat temu, w ramach dziedziny IR, której podstawowym zadaniem jest klasyfikacja elementów do dwóch klas: potencjalnie relewantnych i nirelewantnych dla użytkownika [Rijsbergen 1979] i można je podzielić na następujące kategorie: dokładne dopasowanie, wykorzystanie funkcji podobieństwa i wykorzystanie funkcji odległości do wyznaczenia najbliższego sąsiada oraz klasyfikatory.

### 4.2.1. Dokładne dopasowanie na podstawie słów kluczowych

Dokładne dopasowanie na podstawie słów kluczowych jest stosowane przede wszystkim do wyszukiwania danych i polega w skrócie na wyszukaniu tych elementów, których reprezentacja zawiera te same słowa kluczowe co profil użytkownika [Rijsbergen 1979]. Przydatność tej metody w wyszukiwaniu informacji jest dość ograniczona, gdyż często występują problemy z kontrolą użycia słów kluczowych, synonimią, homonimią, liczbą mnogą, a w przypadku języków fleksyjnych nawet z odmianą przez przypadki, a także występują problemy z kontrolą użycia słów kluczowych złożonych z więcej niż jednego słowa w języku naturalnym.

Obecnie dokładne dopasowanie słów kluczowych, ze względu na wymienione wady, nie jest zbyt często stosowane. Alternatywą jest zastosowanie pewnych modyfikacji pozwalających uniknąć wielu problemów leksykalnych, np. wykorzystanie deskryptorów z tezauryś [Dąbrowski 1978]. Dla przykładu, w systemie rekomendacyjnym *SiteIF* [Magnini 2002], który jest personalnym agentem budującym model użytkownika na podstawie jego przeszłych akcji, tzn. ekstrakcji słów kluczowych z wcześniej odwiedzanych stron webowych, słowa z dokumentów przyporządkowane są do odpowiednich deskryptorów na podstawie WORDNET DOMAINS, wielojęzycznej leksykalnej bazy danych stanowiącej rozszerzenie popularnego WORDNET-u. Po takim procesie indeksowania dokumentów realizowane jest dokładne dopasowanie deskryptorów.



## 4.2.2. Funkcje odległości i podobieństwa

Ponieważ dokładne dopasowanie słów kluczowych ma bardzo ograniczoną przydatność w wyszukiwaniu informacji, a w konsekwencji również ograniczone możliwości rekomendacji określonych elementów, więc zostały opracowane funkcje odległości i podobieństwa, które umożliwiają zniuansowanie zależności między różnymi obiektami. Już około pięćdziesiąt lat temu w wyszukiwaniu informacji zastosowano różnorodne funkcje odległości i podobieństwa, które pozwalały wyszukać dokumenty na podstawie częściowej zgodności profilu użytkownika (bądź jego zapytania) z opisem dokumentu [Salton 1975]. W niniejszym punkcie zostaną przedstawione definicje i własności funkcji odległości i podobieństwa, przedstawione zostaną wybrane przykłady tych funkcji, a także zostanie omówione ich zastosowanie w systemach rekomendacyjnych.

### 4.2.2.1. Definicje i własności funkcji odległości oraz podobieństwa

Funkcje odległości, podobieństwa lub podobieństwa wraz ze zbiorem obiektów  $U$  określają pewną przestrzeń, w której można określać różnego rodzaju zależności, m.in. consensus dla pewnego podzbioru obiektów  $U$  [Nguyen 2002]. Definicje zostały przytoczone za [Soergel 1967] i [Dąbrowski 1978], przy czym funkcja podobieństwa nazywana jest w tych pracach funkcją bliskości, chociaż również i pierwsza nazwa jest używana.

#### Definicja 1. Funkcja odległości

Funkcję  $\delta: U \times U \rightarrow \mathbb{R}_+ \cup \{0\}$  nazywa się funkcją odległości, jeżeli spełniony jest warunek  $\forall_{a,b \in U} \delta(a, a) \leq \delta(a, b)$ .

#### Definicja 2. Funkcja podobieństwa

Funkcję  $\sigma: U \times U \rightarrow \mathbb{R}_+ \cup \{0\}$  nazywamy funkcją podobieństwa, jeżeli  $\forall_{a,b \in U} \sigma(a, a) \geq \sigma(a, b)$ .

#### Definicja 3. Quasi-metryczna funkcja odległości

Jeżeli funkcja odległości spełnia następujący warunek:

$$\forall_{a,b \in U} \delta(a, a) = \delta(b, b) \leq \delta(a, b)$$

oraz warunek symetrii:

$$\forall_{a,b \in U} \delta(a, b) = \delta(b, a),$$

to nazywa się quasi-metryczną funkcją odległości.

#### Definicja 4. Quasi-metryczna funkcja podobieństwa

Jeżeli funkcja podobieństwa spełnia następujący warunek:

$$\forall_{a,b \in U} \sigma(a, a) = \sigma(b, b) \geq \sigma(a, b)$$

oraz warunek symetrii:

$$\forall_{a,b \in U} \sigma(a, b) = \sigma(b, a),$$

to nazywa się quasi-metryczną funkcją odległości.

**Definicja 5. Półmetryczna funkcja odległości**

Jeżeli quasimetryczna funkcja odległości spełnia warunek trójkąta:

$$\forall_{a,b \in U} \delta(a, c) \leq \delta(a, b) + \delta(b, c),$$

to nazywa się ją półmetryczną funkcją odległości.

**Definicja 6. Półmetryczna funkcja podobieństwa**

Jeżeli quasimetryczna funkcja podobieństwa spełnia warunek:

$$\forall_{a,b \in U} \sigma(a, c) \geq \sigma(a, b) \sigma(b, c) / [\sigma(a, b) + \sigma(b, c)],$$

to nazywa się ją półmetryczną funkcją podobieństwa.

**Definicja 7. Metryczna funkcja odległości**

Jeżeli półmetryczna funkcja odległości spełnia warunek:

$$\forall_{a,b \in U \text{ i } a \neq b} \delta(a, a) = \delta(b, b) < \delta(a, b),$$

to nazywa się *metryczną funkcją odległości* (lub w skrócie metryką).

**Definicja 8. Metryczna funkcja podobieństwa**

Jeżeli półmetryczna funkcja podobieństwa spełnia warunek:

$$\forall_{a,b \in U \text{ i } a \neq b} \sigma(a, a) = \sigma(b, b) > \sigma(a, b),$$

to nazywa się *metryczną funkcją podobieństwa* (lub w skrócie metryką).

Nawet pobieżne zapoznanie się z definicjami pozwala nam dostrzec, że istnieje związek między funkcją odległości i funkcją podobieństwa, a odpowiednie warunki wykazują spore analogie. Powoduje to, że każdą zdefiniowaną w punkcie Wybrane funkcje odległości i podobieństwa funkcję odległości można przekształcić w funkcję podobieństwa (i odwrotnie), np. na jeden ze sposobów zaprezentowanych w podanych własnościach dotyczących funkcji odległości i podobieństwa.

**Twierdzenie 1**

Jeżeli  $\delta$  będzie dowolną funkcją odległości przyjmującą wartości dodatnie, to funkcja  $\sigma(a, b) = \max_{x,y \in U} [\delta^n(x, y)] - \delta^n(a, b)$  jeżeli  $\max_{x,y \in U} [\delta(x, y)]$  istnieje i jest skończone oraz  $n = 1, 2, \dots$ , jest funkcją podobieństwa.

**Dowód**

Dowód przeprowadzony zostanie metodą nie wprost.

Założmy, że  $\exists_{a,b \in U} \sigma(a, a) \leq \sigma(a, b)$  oraz przyjmuje wartości dodatnie, tzn. funkcja  $\sigma$  nie jest funkcją podobieństwa, wtedy  $\max_{x,y \in U} [\delta^n(x, y)] - \delta^n(a, a) \leq \max_{x,y \in U} [\delta^n(x, y)] - \delta^n(a, b) \Rightarrow \delta^n(a, a) \geq \delta^n(a, b) \Rightarrow \delta(a, a) \geq \delta(a, b)$ , co jest sprzeczne z założeniem, że  $\delta$  jest funkcją odległości. ■

**Twierdzenie 2**

Jeżeli  $\sigma$  będzie dowolną funkcją podobieństwa przyjmującą wartości dodatnie, to funkcja  $\delta(a, b) = \max_{x, y \in U} [\sigma^n(x, y)] - \sigma^n(a, b)$  jeżeli  $\max_{x, y \in U} [\sigma(x, y)]$  istnieje i jest skończone oraz  $n = 1, 2, \dots$ , jest funkcją odległości.

**Dowód**

Dowód przeprowadzony zostanie metodą nie wprost.

Założmy, że  $\exists_{a, b \in U} \delta(a, a) \geq \delta(a, b)$  oraz  $\delta$  przyjmuje wartości dodatnie, tzn. funkcja nie jest funkcją odległości, wtedy  $\max_{x, y \in U} [\sigma^n(x, y)] - \sigma^n(a, a) \geq \max_{x, y \in U} [\sigma^n(x, y)] - \sigma^n(a, b) \Rightarrow \sigma^n(a, a) \leq \sigma^n(a, b) \Rightarrow \sigma(a, a) \leq \sigma(a, b)$ , co jest sprzeczne z założeniem, że jest funkcją podobieństwa. ■

**Twierdzenie 3**

Jeżeli  $\delta$  będzie dowolną funkcją odległości przyjmującą wartości dodatnie, to funkcja  $\sigma(a, b) = e^{-\delta(a, b)}$  jest funkcją podobieństwa.

**Dowód**

Dowód przeprowadzony zostanie metodą nie wprost.

Założmy, że  $\exists_{a, b \in U} \sigma(a, a) \leq \sigma(a, b)$  oraz  $\sigma$  przyjmuje wartości dodatnie, tzn. funkcja  $\sigma$  nie jest funkcją podobieństwa, wtedy  $e^{-\delta(a, a)} \leq e^{-\delta(a, b)} \Rightarrow e^{\delta(a, a)} \geq e^{\delta(a, b)} \Rightarrow \delta(a, a) \geq \delta(a, b)$  co jest sprzeczne z założeniem, że jest funkcją odległości. ■

**Twierdzenie 4**

Jeżeli  $\sigma$  będzie dowolną funkcją podobieństwa przyjmującą wartości dodatnie, to funkcja  $\delta(a, b) = -\ln \sigma(a, b)$  dla  $\sigma(a, b) \in (0, 1]$ , jest funkcją odległości.

**Dowód**

Dowód przeprowadzony zostanie metodą nie wprost.

Założmy, że  $\exists_{a, b \in U} \delta(a, a) \geq \delta(a, b)$  oraz przyjmuje wartości dodatnie, tzn. funkcja  $\delta$  nie jest funkcją odległości, wtedy  $-\ln \sigma(a, a) \geq -\ln \sigma(a, b)$  dla  $\sigma(a, b) \in (0, 1] \Rightarrow \ln \sigma(a, a) \leq \ln \sigma(a, b)$  dla  $\sigma(a, b) \in (0, 1] \Rightarrow \sigma(a, a) \leq \sigma(a, b)$  dla  $\sigma(a, b) \in (0, 1]$  co jest sprzeczne z założeniem, że jest funkcją podobieństwa. ■

**4.2.2.2. Wybrane funkcje odległości i podobieństwa**

Postać funkcji odległości i podobieństwa w dużej mierze zależy od mikrostruktury zbioru obiektów  $U$  [Nguyen 2001a]. Większość popularnych funkcji odległości i podobieństwa jako mikrostrukturę zbioru wykorzystuje tzw. wektor cech (binarny) lub ważony wektor cech, które omówiono w punkcie 3.3.2.2. Podane funkcje wyznaczają odległość lub podobieństwo pomiędzy dwoma  $n$ -wymiarowymi wektorami  $\mathbf{a}$  i  $\mathbf{b}$ .

**Odległość euklidesowa:**

$$\delta(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \quad (7)$$

**Odległość euklidesowa z ważonymi osiami:**

$$\delta(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{i=1}^n w_i (a_i - b_i)^2}, \quad (8)$$

gdzie  $w_i \in \mathbb{R}^+ \cup \{0\}$ .

**Odległość Hamminga:**

$$\delta(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^n |a_i - b_i|. \quad (9)$$

**Uśredniona odległość Hamminga (odległość Manhattan) [Cho 83]:**

$$\delta(\mathbf{a}, \mathbf{b}) = \frac{1}{n} \sum_{i=1}^n |a_i - b_i|. \quad (10)$$

**Odległość Canbera:**

$$\delta(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^n \frac{|a_i - b_i|}{a_i + b_i}. \quad (11)$$

**Cosinusowa funkcja podobieństwa:**

$$\sigma(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^n a_i b_i}{\sqrt{\sum_{i=1}^n a_i^2 \sum_{i=1}^n b_i^2}}. \quad (12)$$

**Podobieństwo Pearsona [Pearsons 2006]:**

$$\sigma(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^n (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^n (a_i - \bar{a})^2 \sum_{i=1}^n (b_i - \bar{b})^2}}, \quad (13)$$

gdzie  $\bar{a} = \sum_{i=1}^n a_i$  i  $\bar{b} = \sum_{i=1}^n b_i$  są odpowiednio średnimi wartościami dla wektorów  $\mathbf{a}$  i  $\mathbf{b}$ .

Współczynnik ten mierzy liniową zależność między dwiema zmiennymi [Pearsons 2007] i przyjmuje wartości z przedziału  $<-1,1>$ , gdzie wartość  $-1$  oznacza idealną negatywną (odwrotną) korelację, wartość  $0$  brak korelacji, a wartość  $1$  idealną pozytywną korelację. Zakłada się, że wartości wektorów reprezentują zmienne losowe o rozkładzie w przybliżeniu normalnym oraz ich łączny rozkład jest biwariacyjnie normalny (ang. *bivariate normal*). Dla przyspieszenia obliczeń można posłużyć się następującą formułą:

$$\sigma(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^n a_i b_i - \frac{1}{n} \sum_{i=1}^n a_i \sum_{i=1}^n b_i}{\sqrt{\sum_{i=1}^n a_i^2 - \frac{1}{n} \left( \sum_{i=1}^n a_i \right)^2} \sqrt{\sum_{i=1}^n b_i^2 - \frac{1}{n} \left( \sum_{i=1}^n b_i \right)^2}} \quad (14)$$

Na podstawie przeprowadzonych badań opisanych w punkcie 1.2.1 dzięki zastosowaniu szybszej wersji obliczania współczynnika można przyspieszyć obliczenia o około 20%.

### Twierdzenie 5

Formuły (13) i (14) są równoważne.

### Dowód

$$\begin{aligned} & \frac{\sum_{i=1}^n (a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^n (a_i - \bar{a})^2} \sqrt{\sum_{i=1}^n (b_i - \bar{b})^2}} = \frac{\sum_{i=1}^n a_i b_i - \sum_{i=1}^n \bar{a} b_i - \sum_{i=1}^n a_i \bar{b} + \sum_{i=1}^n \bar{a} \bar{b}}{\sqrt{\sum_{i=1}^n (a_i - \bar{a})^2} \sqrt{\sum_{i=1}^n (b_i - \bar{b})^2}} \\ & = \frac{\sum_{i=1}^n a_i b_i - \bar{a} \sum_{i=1}^n b_i - \bar{b} \sum_{i=1}^n a_i + n \frac{1}{n} \sum_{i=1}^n a_i \frac{1}{n} \sum_{i=1}^n b_i}{\sqrt{\sum_{i=1}^n (a_i^2 - 2\bar{a}a_i + \bar{a}^2)} \sqrt{\sum_{i=1}^n (b_i^2 - 2\bar{b}b_i + \bar{b}^2)}} \\ & = \frac{\sum_{i=1}^n a_i b_i - \frac{1}{n} \sum_{i=1}^n a_i \sum_{i=1}^n b_i - \frac{1}{n} \sum_{i=1}^n b_i \sum_{i=1}^n a_i + \frac{1}{n} \sum_{i=1}^n a_i \sum_{i=1}^n b_i}{\sqrt{\sum_{i=1}^n a_i^2 - 2 \frac{1}{n} \sum_{i=1}^n a_i \sum_{i=1}^n a_i + n \frac{1}{n} \sum_{i=1}^n a_i \frac{1}{n} \sum_{i=1}^n a_i} \sqrt{\sum_{i=1}^n b_i^2 - 2 \frac{1}{n} \sum_{i=1}^n b_i \sum_{i=1}^n b_i + n \frac{1}{n} \sum_{i=1}^n b_i \frac{1}{n} \sum_{i=1}^n b_i}} \end{aligned}$$

$$\begin{aligned}
&= \frac{\sum_{i=1}^n a_i b_i - \frac{1}{n} \sum_{i=1}^n a_i \sum_{i=1}^n b_i - \frac{1}{n} \sum_{i=1}^n b_i \sum_{i=1}^n a_i + \frac{1}{n} \sum_{i=1}^n a_i \sum_{i=1}^n b_i}{\sqrt{\sum_{i=1}^n a_i^2 - 2 \frac{1}{n} \sum_{i=1}^n a_i \sum_{i=1}^n a_i + n \frac{1}{n} \sum_{i=1}^n a_i \frac{1}{n} \sum_{i=1}^n a_i} \sqrt{\sum_{i=1}^n b_i^2 - 2 \frac{1}{n} \sum_{i=1}^n b_i \sum_{i=1}^n b_i + n \frac{1}{n} \sum_{i=1}^n b_i \frac{1}{n} \sum_{i=1}^n b_i}} \\
&= \frac{\sum_{i=1}^n a_i b_i - \frac{1}{n} \sum_{i=1}^n a_i \sum_{i=1}^n b_i}{\sqrt{\sum_{i=1}^n a_i^2 - \frac{1}{n} \left( \sum_{i=1}^n a_i \right)^2} \sqrt{\sum_{i=1}^n b_i^2 - \frac{1}{n} \left( \sum_{i=1}^n b_i \right)^2}}
\end{aligned}$$

**Podobieństwo P1 [Dąbrowski 1978]:**

$$\sigma(\mathbf{a}, \mathbf{b}) = \frac{1}{n} \left( c_1 \sum_{i=1}^n \min(a_i, b_i) + c_2 \sum_{i=1}^n \min(\bar{a}_i, \bar{b}_i) \right), \quad (15)$$

gdzie:  $c_1, c_2 = \{0, 1, \dots, n\}$ .

### Edycyjne miary podobieństwa

Dla odmiennych mikrostruktur stosuje się również inne miary odległości i podobieństwa. W przypadku, gdy elementy zawierają ciągi znaków wówczas najczęściej stosuje się miary edycyjne [Cohen 2003], które mierzą odległość między ciągami  $s$  i  $t$  jako koszt najlepszej sekwencji operacji edycyjnych przekształcających  $s$  w  $t$ . Do typowych operacji edycyjnych należą: wstawienie, skasowanie oraz zastąpienie znaku, przy czym każda operacja ma przypisany koszt. W przypadku odległości Levensteina jednakowy koszt przypisywany jest do każdej operacji. Przykładem bardziej złożonej miary jest funkcja odległości Mongera i Elkana, będąca afinicznym wariantem odległości Smitha i Watermana [Cohen 2003] z określonymi parametrami kosztu i znormalizowanej do przedziału  $[0, 1]$ .

### Podobieństwo Jaro

Przykładem funkcji podobieństwa między dwoma ciągami, jednak nie opartej na operacjach edycyjnych, jest np. funkcja Jaro, która bierze pod uwagę liczbę i kolejność wspólnych znaków [Cohen 2003]. Niech  $s = a_1 \dots a_K$  oraz  $t = b_1 \dots b_L$  oraz  $a_i$  z ciągu  $s$  jest **wspólne** z ciągiem  $t$ , jeżeli istnieje takie  $b_j$  z ciągu  $t$  równe  $a_i$ , dla którego zachodzi dodatkowo następująca zależność:  $i - H \leq j \leq i + H$ , gdzie  $H = \min(|s|, |t|)/2$ . Niech  $s' = a'_1 \dots a'_K$  będzie ciągiem wspólnych znaków z  $t$  (kolejność z  $s$ ) oraz analogicznie  $t' = b'_1 \dots b'_L$  będzie ciągiem wspólnych znaków  $t$  z  $s$ . Wtedy można zdefiniować **transpozycję** dla  $s'$  i  $t'$  jako występowanie takiego  $i = 1, 2, \dots, K$ , dla którego zachodzi  $a'_i \neq b'_i$ , oznaczmy następnie  $T$  jako liczba transpozycji dla  $s'$  i  $t'$ . Wtedy funkcja podobieństwa Jaro wynosi:

$$\beta(s, t) = \frac{1}{3} \left( \frac{|s'|}{|s|} + \frac{|t'|}{|t|} + \frac{|s'| - T}{|s'|} \right). \quad (16)$$



### Podobieństwo Jaccarda

Ciągi mogą składać się nie tylko z pojedynczych znaków, ale także z bardziej złożonych elementów, takich jak słowa, terminy czy pewne znaczniki (ang. *tokens*). W tym przypadku można zastosować np. podobieństwo Jaccarda między zbiorami słów  $S$  i  $T$  wyrażone wzorem:

$$\beta(s, t) = \frac{|S \cap T|}{|S \cup T|} \quad (17)$$

### Podobieństwo TFIDF [Salton 1983]:

Miara podobieństwa TFIDF (ang. *Term Frequency Inverted Document Frequency*) została opracowana i szeroko wykorzystywana na gruncie dziedziny IR. Służy ona z jednej strony do indeksowania, jak również wyznaczania podobieństw między profilami, które powstały w rezultacie realizacji procesu indeksowania. Niech  $TF_{w,S}$  będzie częstością występowania słowa  $w$  w zbiorze  $S$ ,  $N$  będzie liczebnością całej rozważanej kolekcji (np. dokumentów),  $IDF_w$  będzie odwrotnością częstości występowania  $w$  w całej kolekcji,  $V'(w, S) = \log(TF_{w,S} + 1) \log(IDF_w)$ ,  $V(w, S) = \frac{V'(w, S)}{\sqrt{\sum_{w' \in S} (V'(w', S))^2}}$ , wtedy podobieństwo TFIDF wyraża się następującym wzorem:

$$\beta(S, T) = \sum_{w \in S \cap T} V(w, S) V(w, T) \quad (18)$$

### Ważony współczynnik $\kappa$ Chena [Chen 2004]:

Współczynnik ten wykorzystuje się do wyznaczenia stopnia zgodności opinii między dwoma obserwatorami [Viera 2005]. Można go zastosować, gdy wektory  $\mathbf{a}$  i  $\mathbf{b}$  zawierają skończoną liczbę wartości (np. ocen lub rang elementów):

$$\kappa = \frac{1}{m} \sum_{k=1}^g \sum_{l=1}^g w_{k,l} f_{k,l}, \quad (19)$$

gdzie:  $g$  – liczba wartości ocen,

$m$  – liczba elementów, dla których jednocześnie w wektorach  $\mathbf{a}$  i  $\mathbf{b}$  występują niepuste wartości,

$f_{k,l}$  – współczynnik macierzy  $f$ , która powstała na podstawie wyznaczenia liczby elementów mających w wektorze  $\mathbf{a}$   $k$ -tą ocenę i  $l$ -tą ocenę w wektorze  $\mathbf{b}$ ,

$w_{k,l}$  – wartość wagowa dla współczynnika o indeksie  $k, l$ , współczynnik ten wyznacza się na podstawie wzoru:

$$w_{k,l} = 1 - \frac{|k-l|}{g-1}, \text{ wtedy mówimy o wersji liniowej,}$$

$$w_{k,l} = 1 - \frac{(k-l)^2}{(g-1)^2}, \text{ wtedy mówimy o wersji kwadratowej współczynnika}$$

[Viera 2005].

### Współczynnik korelacji $\tau$ Kendalla [Abdi 2007]:

Można go zastosować podobnie, jak w przypadku współczynnika  $\kappa$  Chena [Chen 2004], gdy wektory  $\mathbf{a}$  i  $\mathbf{b}$  zawierają skończoną liczbę wartości (np. ocen lub rang elementów):

$$\tau(\mathbf{a}, \mathbf{b}) = \frac{\frac{1}{2}n(n-1) - P}{\frac{1}{2}n(n-1)} = 1 - \frac{2P}{n(n-1)}, \quad (20)$$

gdzie:  $n$  – liczba elementów w wektorach  $\mathbf{a}$  i  $\mathbf{b}$  o tej samej długości,

$P$  – licznosc zbioru będącego różnicą symetryczną zbiorów par porządków (ang. *ordered pairs*) dla wektorów rangi  $\mathbf{a}$  i  $\mathbf{b}$ , przy czym liczba par porządków w zbiorze uporządkowanym wynosi  $\frac{1}{2}n(n-1)$ .

Gdy wektory  $\mathbf{a}$  i  $\mathbf{b}$  są wektorami ocen na pewnej skali (np. od 1 do 5, tak jak w przypadku zbioru *MovieLens*), wówczas wartość  $P$  wyznaczana jest w następujący sposób: na podstawie  $n$ -elementowych wektorów ocen  $\mathbf{a}$  i  $\mathbf{b}$  tworzymy zbiory par porządków indeksów elementów z wektorów  $\mathbf{a}$  i  $\mathbf{b}$ , a następnie tworzymy na ich podstawie uporządkowane zbiory binarne, przyjmujące wartość 1, jeżeli element o niższym indeksie ma większą lub równą ocenę niż element o wyższym indeksie. Wartość  $P$  wyznaczana jest jako liczba różnic między zbiorami binarnymi o długości  $\frac{1}{2}n(n-1)$  tworzonymi na podstawie wektorów  $\mathbf{a}$  i  $\mathbf{b}$ . Współczynnik korelacji  $\tau$  Kendalla ma następujące własności [Abdi 2007]:

- współczynnik przyjmuje wartości od  $-1$  dla zupełnie różnych wektorów rang lub ocen oraz  $1$  dla tych samych uporządkowań,
- dla pozostałych wartości między  $-1$  a  $1$ , rosnąca wartość współczynnika oznacza wzrastającą zgodność między rankingami; wartość współczynnika wynosząca  $0$  oznacza zupełną niezależność.

#### 4.2.2.3. Zastosowanie funkcji odległości i podobieństwa w systemach rekomendacyjnych

Zaprezentowane w poprzednim punkcie funkcje odległości i podobieństwa z dużym powodzeniem były wykorzystywane w wielu aplikacjach systemów rekomendacyjnych. Zaprezentowane funkcje podobieństwa, szczególnie podobieństwo cosinusowe, TFIDF i podobieństwo Jaccarda były również wykorzystywane w wielu systemach wyszukiwania informacji, m.in. w systemie SMART [Salton 1975], [Rijsbergen 1978], [Salton 1983]. Znalazły one również zastosowanie w rozlicznych systemach rekomendacyjnych w celu określenia dopasowania profilu do elementu. Wiele funkcji podobieństwa: Pearsona,  $\tau$  Kendalla i ważony współczynnik  $\kappa$  Chena, wykorzystywanych jest raczej do określenia podobieństwa między profilami lub elementami, co szerzej zostanie omówione w punkcie 4.3. W pracy [Montaner 2003], jako przykłady wykorzystania podobieństwa kosinusowego wymienia się następujące systemy: *Amalthaea*, *Anatagonomy*, *Fab*, *Kra-*

*katoa Chronicle*, *Let's Browse*, *Letizia*, *MovieLens*, *NewsWeeder*, *NewT*, *Syskill & Webert*, *Webmate*, *WebWatcher*; czyli przynajmniej 1/3 z analizowanych w tej pracy systemów. Pozostałe metody były wykorzystywane o wiele rzadziej. Funkcja TFIDF jest również bardzo popularna w wyszukiwaniu informacji, stąd jest również wykorzystywana w systemach rekomendacyjnych, jednak przede wszystkim do uczenia profilu; podobieństwo między tak utworzonymi profilami wyznaczane jest przez podobieństwo cosinusowe. Takie podejście realizowane jest m.in. w następujących systemach rekomendacyjnych [Montaner 2003]: *Amalthaea*, *Anatagonomy*, *Fab*, *Krakatoa Chronicle*, *Let's Browse*, *Letizia*, *MovieLens*, *NewsWeeder*, *NewT*, *Syskill & Webert*, *Webmate*, *WebWatcher* oraz *FWUM* [Symeonidis 2007]. Jednak również i inne funkcje podobieństwa są nadal w użyciu, np. funkcja Jaccarda wykorzystywana jest w systemie *Quambo* [Elliott 2008]. Funkcję tę można zastosować do określenia dopasowania między poszczególnymi elementami, np. w eksperymentach z baza danych zawierającą opisy produktów multimedialnych przeznaczonych dla telefonów komórkowych, takich jak logosy, dzwonki i gry. Zostaną przedstawione eksperymenty dotyczące czasu obliczeń dla 1 miliona ocen zawartych w bazie *MovieLens* [GroupLens 2008], w dalszej części pracy (w punkcie 4.3.1) zostaną przedstawione wyniki badań nad efektywnością rekomendacji z wykorzystania tych miar.

### 4.2.3. Klasyfikacja

Klasyfikacja, historycznie rzecz biorąc, była stosowana przede wszystkim w tzw. klasyfikacji bibliotecznej. W późniejszym okresie znalazła zastosowanie w dziedzinie IR [Rijsbergen 1978], gdzie określana jest jako proces, w którym konstruuje się system klasyfikacyjny lub jako rezultat tego procesu (przy czym autor zdecydowanie wyklucza spotykane czasem w literaturze przedmiotu utożsamianie klasyfikacji z indeksowaniem dokumentów). W dziedzinie IR stosowano klasyfikację w grupowaniu (ang. *clustering*) słów kluczowych i grupowaniu dokumentów [Salton 1975].

W dziedzinie systemów rekomendacyjnych (RS) klasyfikacja jako metoda dopasowania profilu do elementu rozumiana jest jako podział zbioru elementów na podzbiory elementów rekomendowanych i nierekomendowanych, interesujących i nieinteresujących, lub relewantnych, niereklewantnych i nieokreślonych. Stosuje się tu wiele różnych metod, w których profil wykorzystywany jest jako klasyfikator dla: sieci neuronowej, drzewa decyzyjnego, reguł indukcyjnych, wnioskowania rozmytego i sieci bayesowskiej [Miyahara 2000].

Podane metody klasyfikacji stosowane są w przypadku wykorzystania następujących metod filtrowania: CBF, DF i CBR. Dla przykładu, w systemie *Megane*, opisanym dokładnie w rozdziale 5, na podstawie profilu użytkownika zawierającego dane demograficzne użytkownicy przydzielani są do klasy, do której centroidu profil ma najmniejszą odległość. Każda klasa determinuje bezpośrednio ustawienia interfejsu użytkownika. W innym systemie rekomendującym przepisy kulinarne *Kucharek* [Sobecki 2006a], klasyfikacja użytkowników, również na podstawie profili zawierających dane demograficzne, realizowana jest z wykorzystaniem wnioskowania rozmytego.

### 4.3. Dopasowanie między profilami

Dopasowanie między profilami wykorzystywane jest przede wszystkim w filtrowaniu kolaboratywnym (CF) realizowanym w następujących trzech krokach [Montaner 2003]: znajdź podobnych użytkowników, stwórz sąsiedztwo oraz wyznacz predykcję na podstawie określonego sąsiedztwa.

#### 4.3.1. Znajdowanie podobnych użytkowników

Znajdowanie podobnych użytkowników odbywa się najczęściej jedną z trzech podstawowych metod: najbliższy sąsiad, grupowanie i klasyfikatory. Metody te wykorzystują różne funkcje podobieństwa lub rzadziej odległości przedstawione w punkcie 4.2.2.

Metoda najbliższego sąsiada dla określonego użytkownika polega na wybraniu spośród kolekcji użytkowników reprezentowanych przez odpowiednie profile pewnej ich liczby, które to profile spełniają dodatkowo pewne dodatkowe warunki. Tymi warunkami są najczęściej: minimalna wartość odległości lub maksymalna wartość podobieństwa.

Podstawową metodą wyznaczenia podobnych użytkowników jest zastosowanie jednej z przedstawionych w punkcie 4.2.2 funkcji podobieństwa do wyznaczenia zgodności między profilami różnych użytkowników. Najbardziej popularnymi funkcjami są funkcje cosinusowa i funkcja Pearsona [Montaner 2003], [Nguyen 2007]. Funkcja cosinusowa była wykorzystywana w następujących systemach: *Anatagonomy*, *Fab*, *Krakatoa Chronicle*, *MovieLens*, *NewsWeeder* i *WebWatcher*, natomiast funkcję Pearsona zastosowano np. w systemach rekomendacyjnych takich jak: *GroupLens*, *Ringo/Fire-Fly*, *Smart Radio*, *WebSell* oraz *The Movie* [Nguyen 2007]. Funkcja  $\kappa$  została zastosowana w systemie *Reporter* [Sobecki 2007a].

Oczywiście, oceniając przydatność określonej funkcji podobieństwa w systemie rekomendacyjnym, należy wziąć pod uwagę przede wszystkim efektywność rekomendacji, którą można uzyskać wykorzystując odpowiednią funkcję. Porównanie efektywności rekomendacji w zależności od zastosowanej funkcji podobieństwa przedstawiono w dalszej części pracy w rozdziale 4.4. Jednak oprócz efektywności rekomendacji bardzo istotną cechą każdej funkcji podobieństwa jest również czas obliczeń potrzebny do jej wyznaczenia. Badania porównawcze czasu wyznaczenia macierzy podobieństw dla wybranych metod podobieństwa zostały przeprowadzone dla bazy *MovieLens* zawierającej 1 milion ocen 3952 filmów wykonanych przez 6040 użytkowników [Gouplens 2008].

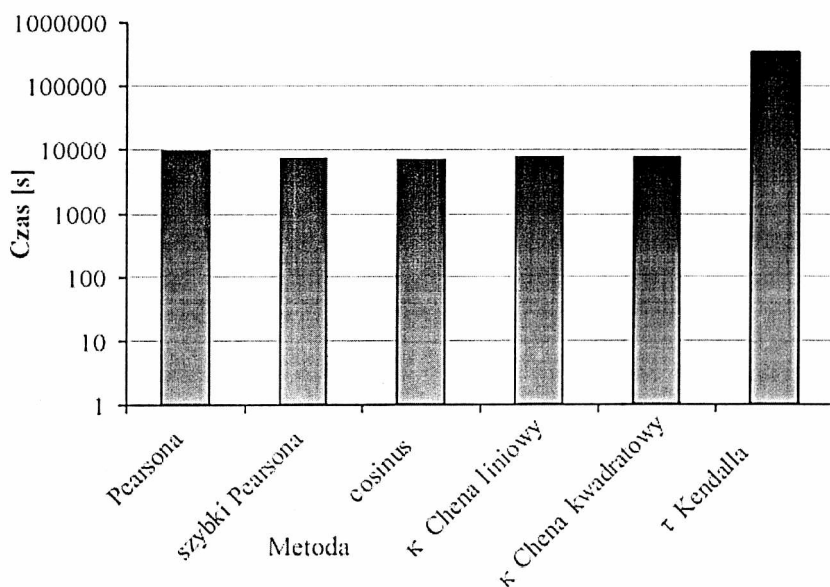
Badania zostały przeprowadzone na komputerze o następującej konfiguracji: procesor 3 GHz Dual Core, RAM 3 GB, system MS Windows XP Professional. W ramach badań przeprowadzono wyznaczenie macierzy podobieństw z wykorzystaniem następujących funkcji podobieństw (współczynników korelacji), w nawiasie podano czas realizacji: Pearsona (9529,63 s – 158 min 50 s), szybki Pearsona (7536,99 s – 125 min 37 s), cosinus (6843,42 s – 114 min 3 s),  $\kappa$  Chena liniowy (7666,03 s – 127 min 46 s),  $\kappa$  Chena kwadratowy (7703,82 s – 128 min 24 s),  $\tau$  Kendalla (353936,66 s – 5898 min 57 s). Na rysunku 11 porównano czas obliczenia dla tych funkcji podobieństwa na skali logarytmicznej.

Można zauważyć, że dla większości metod, na zastosowanej platformie, obliczenie współczynnika podobieństwa dla pojedynczego użytkownika trwa od 1 do 1,5 sekund. Teoretycznie czas ten jest na granicy akceptowalności, jednak w realnie działającym systemie, gdzie jednocześnie mogą być obsługiwane nawet setki lub tysiące użytkowników oraz gdy liczba zarejestrowanych użytkowników sięga setek tysięcy wówczas wyznaczanie podobieństwa tą metodą raczej nie może być stosowane w trybie on-line. Obliczenia te dają się jednak łatwo zrównoleglić, stąd stosując klaster komputerów lub grid można obliczenia te zdecydowanie przyspieszyć. Tym nie mniej w przypadku wielu dziesiątek tysięcy użytkowników należy posłużyć się raczej grupowaniem użytkowników off-line oraz znalezieniem najbliższego centroida dla określonego (nawet nowego użytkownika) on-line.

W ramach niniejszych badań przeprowadzono również obliczenie macierzy podobieństw metodą  $\tau$  Kendalla [Abdi 2007]. Metoda ta jednak okazała się bardzo wymagająca obliczeniowo, gdyż czas obliczeń uzyskany na tym samym zestawie komputerowym był około 50 razy większy od uzyskanego pozostałymi metodami.

Innym podejściem bazującym na korelacji jest podejście oparte na miarach entropii niepewności wykorzystujące techniki prawdopodobieństwa warunkowego w celu mierzenia spadku entropii aktualnego użytkownika wynikająca ze znajomości ocen innego użytkownika [Montaner 2003]. W literaturze można jednak znaleźć porównanie tej metody do metody wyznaczania predykcji np. ze współczynnikiem korelacji Pearsona, z którego wynika, iż metoda entropii jest mniej efektywna [Herlocker 1999].

Oprócz metody najbliższego sąsiedztwa, podobnych użytkowników można znaleźć przez grupowanie i klasyfikację, które zostały szczegółowo opisane odpowiednio w punktach 3.3.4.2 oraz 3.3.4.3 i 4.2.3, przy czym w punktach tych przedstawiono grupowanie



Rys. 11. Czas obliczeń macierzy podobieństw dla zbioru MovieLens 1 milion wybranymi metodami

i klasyfikację profili użytkowników lub elementów, co oczywiście na realizację samej metody nie ma wielkiego wpływu. Wykorzystanie metody grupowania z zastosowaniem algorytmu Datolli oraz klasyfikacji na podstawie wyznaczonych centroidów zostało przedstawione w pracach [Nguyen 2003a] i [Sobecki 2004b].

### 4.3.2. Utworzenie sąsiedztwa

Przy tworzeniu sąsiedztwa można posłużyć się generalnie dwoma technikami [Montaner 2003]: progu korelacji (ang. *correlation thresholding*) i najlepszych  $n$ -sąsiadów (ang. *best n-neighbours*). W niektórych zastosowaniach do sąsiedztwa zalicza się wszystkich użytkowników, jednak wpływ ich oceny na predykcję zależy proporcjonalnie od wartości korelacji.

Zastosowanie progu korelacji polega na określeniu pewnej jej wartości, od której zaliczać się będzie jednego użytkownika do sąsiedztwa drugiego. Jest to metoda dość prosta i stosowana w wielu systemach, np. w *The Movie* [Nguyen 2007]; może ona powodować określone trudności polegające na tym, że przy odpowiednio wysokich progach, możemy otrzymać bardzo nieliczne sąsiedztwo, a w skrajnych przypadkach nawet puste, co spowoduje, że nie będziemy w stanie dostarczyć predykcji danego elementu w przypadku, gdy żaden z członków sąsiedztwa nie ocenił elementu, dla którego chcemy wyznaczyć predykcję.

Druga metoda, najlepszych  $n$ -sąsiadów, nie niesie za sobą zagrożenia, iż nie znajdziemy odpowiedniej liczby sąsiadów, jednak sąsiedztwo to może zawierać użytkowników bardzo słabo skorelowanych. Uwzględnienie bardzo słabo skorelowanych użytkowników na równi z silnie skorelowanymi może przyczynić się do otrzymania słabej predykcji. W takim przypadku należy zastosować rozwiązanie podobne jak w przypadku, gdy do sąsiedztwa zalicza się wszystkich użytkowników.

Dodatkowym warunkiem tworzenia sąsiedztwa może być liczba wspólnie ocenianych elementów. W tym wypadku zaliczani są do sąsiedztwa jedynie tacy użytkownicy, którzy ocenili przynajmniej określoną liczbę elementów. Takie ograniczenie, które może zostać zastosowane zarówno do pierwszej, jak i drugiej metody tworzenia sąsiedztwa, może powodować ograniczenie liczby użytkowników należących do sąsiedztwa, a co za tym idzie problemy z wyznaczeniem predykcji.

### 4.3.3. Wyznaczenie predykcji na podstawie wybranego sąsiedztwa

Wyznaczenie najbliższego sąsiedztwa dla określonego użytkownika jest dopiero pierwszym krokiem wyznaczenia rekomendacji. Kolejnym krokiem jest wyznaczenie predykcji np. oceny określonego elementu dla danego użytkownika. Podstawy teoretyczne dla wyznaczenia tych wartości można znaleźć na gruncie teorii consensusu [Day 1988], [Nguyen 2008]. W większości prezentowanych przykładów wyznaczania predykcji zakłada się, że dysponujemy ocenami innych użytkowników wyrażonymi najczęściej przez liczbę należącą do zbioru skończonego  $\{1, 2, \dots, 5\}$ , tak jak w przypadku zbiorów *MovieLens* [GroupLens 2008].

Jedną z podstawowych funkcji pozwalającą wyznaczyć predykcję, będącą consensusem między poszczególnymi opiniami występującymi w określonej grupie, jest mediana [McMorris 1995], czyli predykcja  $p_{a,j}$  dla użytkownika  $u_a$  i elementu  $j$  wyrażona jest następującym wzorem:

$$p_{a,j} = \mu_{1/2}(r_j^a), \quad (21)$$

gdzie  $r_j^a$  – zmienna przyjmująca wartości oceny elementu  $j$  przez wszystkich użytkowników należących do wybranego sąsiedztwa użytkownika  $u_a$ ,

$\mu_{1/2}$  – mediana, czyli wartość środkowa lub drugi kwartył dla zmiennej  $r_j^a$ .

Kolejną bardzo prostą, aczkolwiek bardzo często stosowaną metodą wyznaczenia predykcji jest zastosowanie średniej arytmetycznej ocen sąsiadów

$$p_{a,j} = \frac{1}{n} \sum_{i=1}^n r_{i,j}, \quad (22)$$

gdzie:  $n$  – liczba sąsiadów danego użytkownika  $a$ ,

$r_{i,j}$  – ocena użytkownika  $i$ , którą przyznał elementowi  $j$ .

Efektywne wykorzystanie średniej arytmetycznej do predykcji ocen wymaga, aby rozkład ocen dla wszystkich użytkowników był prawie identyczny, co praktycznie nie jest możliwe, gdyż jedni użytkownicy mogą mieć tendencję do nadawania wysokich, a inni do nadawania niskich ocen.

Średnia może być z powodzeniem wykorzystywana w przypadku, gdy nie mamy informacji o użytkowniku pozwalającej wyznaczyć sąsiedztwo i wyznaczamy średnią ocenę z całej populacji użytkowników [Herlocker 1999]. Pewną modyfikacją wykorzystania średniej jest zastosowanie pewnych wag (np. wartości funkcji podobieństwa), które modyfikują wagność określonej oceny:

$$p_{a,j} = \frac{\sum_{i=1}^n w_{a,i} r_{i,j}}{\sum_{i=1}^n w_{a,i}}, \quad (23)$$

gdzie:  $w_{a,i}$  – waga zależności między użytkownikiem  $a$  oraz  $i$ , np. podobieństwo Pearsona między nimi.

W literaturze można jednak częściej znaleźć przykłady wykorzystania predykcji, uwzględniającej średnią dla użytkownika i pewne dostosowanie [Papagelis 2005]. Pierwszym, najprostszym tego typu wzorem jest wzór, który uwzględnia odchylenie od średniej ocen sąsiadów:

$$p_{a,j} = \bar{r}_a + \frac{1}{n} \sum_{i=1}^n (r_{i,j} - \bar{r}_i), \quad (24)$$

gdzie:  $\bar{r}_a$  – średnia ocen dla danego użytkownika  $a$ ,

$\bar{r}_i$  – średnia ocena dla użytkownika  $i$ .



Kolejne wzory mogą uwzględniać odchylenie od średniej ważonej ocen [Herlocker 1999]:

$$p_{a,j} = \bar{r}_a + \frac{\sum_{i=1}^n w_{a,i} (r_{i,j} - \bar{r}_i)}{\sum_{i=1}^n w_{a,i}}, \quad (25)$$

bądź również dodatkową normalizację wyrażoną odchyleniami standardowymi dla wartości ocen użytkownika  $a$  i poszczególnych użytkowników z jego sąsiedztwa [Herlocker 1999]:

$$p_{a,j} = \bar{r}_a + \sigma_a \frac{\sum_{i=1}^n w_{a,i} \frac{(r_{i,j} - \bar{r}_i)}{\sigma_i}}{\sum_{i=1}^n w_{a,i}}, \quad (26)$$

gdzie:  $\sigma_a$  – odchylenie standardowe dla wartości ocen użytkownika  $a$ ,

$\sigma_i$  – odchylenie standardowe dla wartości ocen użytkownika  $i$ .

Oprócz różnych modyfikacji wzorów wyznaczania predykcji rekomendowanej oceny, można modyfikować same wagi korelacji. Modyfikacja taka może być zrealizowana przez zastosowanie tzw. współczynnika istotności (ang. *significance weight*), który modyfikuje wagę  $w_{a,i}$  przez zastosowanie prostego współczynnika  $m/k$  dla  $m < k$  i równy 1 w przeciwnym razie, gdzie  $k$  jest stałą reprezentującą wystarczającą liczbę wspólnie ocenionych filmów, a  $m$  liczbę wspólnie ocenionych filmów przez użytkowników  $a$  oraz  $i$  [Herlocker 1999].

## 4.4. Metody oceny efektywności systemów rekomendacyjnych

Ocena efektywności systemów rekomendacyjnych jest elementem niezbędnym w całym procesie ich tworzenia, implementowania i doskonalenia. Bez miarodajnych metod wyznaczania efektywności systemów rekomendacyjnych nie da się osiągnąć postępu w pracach nad ich rozwojem. W literaturze, począwszy od publikacji z dziedziny IR [Salton 1975] po publikacje z zakresu systemów rekomendacyjnych [Herlocker 2001] można znaleźć dziesiątki różnych parametrów ilościowych wraz z odpowiednią techniką ich wyznaczania służącą do oceny efektywności systemów rekomendacyjnych.

Potrzeba badania efektywności systemów rekomendacyjnych, podobnie jak systemów IR, wynika przede wszystkim z przyczyn społecznych, ekonomicznych i rozwojowych [Rijsbergen 1979]. Ewaluacja powinna dostarczyć zarówno użytkownikom, dysponentom, jak i ich twórcom, informacji umożliwiających udzielenie odpowiedzi na następu-

jące pytania: czy użytkownicy będą chcieli używać systemu (aspekt społeczny), czy jego używanie będzie opłacalne (zarówno dla użytkowników, jak i przede wszystkim dla dysponentów) oraz czy zastosowane nowe algorytmy rekomendacji zapewniają ulepszenia w porównaniu z wcześniej znanymi i używanymi metodami (aspekt rozwojowy) [Sobecki 1999a].

Parametry, które podlegają badaniu efektywności wyszukiwania informacji zostały określone już w 1966 przez Cleverdon [Rijsbergen 1979] i są one, w tej lub innej formie, brane pod uwagę przy współczesnym badaniu systemów rekomendacyjnych. Cleverdon wyróżnił następujących sześć mierzalnych wielkości:

- pokrycie (ang. *coverage*) kolekcji, czyli stopień, do którego system zawiera relewantne informacje,
- opóźnienie (ang. *time lag*), ilość czasu upływająca od zadania pytania do otrzymania odpowiedzi,
- forma prezentacji (ang. *presentation form*) informacji wyjściowych,
- nakład pracy (ang. *effort*) wydatkowany przez użytkownika na otrzymanie odpowiedzi systemu,
- kompletność odpowiedzi (ang. *recall*), czyli stosunek relewantnych i wyszukanych elementów, do wszystkich relewantnych do danego pytania elementów dostępnych w bazie,
- dokładność odpowiedzi (ang. *precision*), czyli stosunek relewantnych i wyszukanych elementów, do wszystkich wyszukanych elementów na zadane pytanie.

Uważa się, że cztery pierwsze parametry są łatwe do wyznaczenia, natomiast obliczanie dwóch ostatnich, tj. kompletności i dokładności może nastęrczać więcej trudności, stanowią one jednak w sposób istotny o skuteczności systemu wyszukiwania informacji, gdyż oferują miary dostarczania relewantnej informacji z jednoczesnym niedostarczaniem informacji nirelewantnej. Bardzo często zakłada się, że te dwa parametry są wystarczające do określenia skuteczności systemu, gdyż są najczęściej używaną i powszechnie zrozumiałą parą miar [Rijsbergen 1979], przy czym należy tutaj dodać, że mogą wystąpić pewne problemy z rozumieniem dość podstawowego pojęcia relewancji.

Wymienione miary oceny efektywności mogą być z powodzeniem stosowane na gruncie systemów rekomendacyjnych, jednak ze względu na różnice między dziedzinami IR a RS odmienne są akcenty i istotność pewnych miar. Wraz z rozwojem technologii pojawiły się także nowe miary i ich ujęcie [Setten 2005]:

- dokładność predykcji (ang. *prediction accuracy*), odpowiednik dokładności odpowiedzi,
- stabilność predykcji (ang. *prediction stability*), która pozwala użytkownikowi na określenie poziomu zaufania do otrzymywanych predykcji,
- szybkość predykcji (ang. *prediction speed*), odpowiednik opóźnienia,
- siła objaśniania (ang. *explanatory power*) systemu rekomendacyjnego, zaproponowana w pracy [Herlocker 2004], a wywodząca się z cech systemów ekspertowych, które powinny objaśniać sposób dojścia do swoich konkluzji [Zieliński 2000].

Ocena efektywności może być zadaniem dość trudnym z kilku powodów. Przede wszystkim porównanie metod rekomendacji wymaga, aby zostało ono przeprowadzone na podstawie tego samego zbioru danych wejściowych. Efektywność wielu metod rekomendacji, szczególnie kolaboratywnych, w dużej mierze zależy również od charakteru danych, czyli inaczej mogą sprawdzać się metody rekomendacji, gdzie użytkowników jest o wiele więcej niż rekomendowanych elementów (np. dane *MovieLens* [GroupLens 2008] zawierające 1 milion ocen 3952 filmów wykonanych przez 6040 użytkowników), niż systemy rekomendacyjne, w których występuje o wiele więcej rekomendowanych elementów niż użytkowników, np. systemy wspomagające wyszukiwanie w sieci Internet.

Kolejnymi problemami badania efektywności są: przecenianie znaczenia dokładności predykcji z jednoczesnym pomijaniem znaczenia stabilności predykcji oraz brak porównywalności wykorzystanych miar, np. takich jak MAE dla systemów rekomendacyjnych stosujących różne skale ocen [Herlocker 2004].

W niniejszym rozdziale przedstawione zostaną miary wywodzące się z dziedziny IR bazujące na pojęciu relewancji, miary stosowane często w dziedzinie RS wykorzystujące skalę ocen oraz przykładowe wyniki badań wybranych metod rekomendacji.

#### 4.4.1. Miary efektywności wywodzące się z IR

Główne miary efektywności wywodzące się z dziedziny wyszukiwania informacji (IR) – kompletność i dokładność – bazują na podstawowym pojęciu relewancji. Relewanca definiowana jest jako odpowiedniość między zapytaniem zadany przez użytkownika a dokumentem [Dąbrowski 1978]. Miara ta jest miarą subiektywną [Rijsbergen 1979], tzn. różni użytkownicy zadający to samo pytanie mogą ocenić relewancję określonych dokumentów w różny sposób. Miara relewancji, która jest miarą obiektywną nosi nazwę relewancji logicznej. Należy zwrócić jednak uwagę, że oprócz pojęcia relewancji, czasem wykorzystuje się pojęcie pertynencji, które definiowane jest jako odpowiedniość rozumiana nie w odniesieniu do zapytania użytkownika, ale raczej w odniesieniu do jego aktualnych potrzeb [Dąbrowski 1978]. Różnica między relewancją a pertynencją jest dość istotna, jednak szczególnie przez wielu użytkowników systemów rekomendacyjnych może nie być dostrzegana, stąd otrzymując oceny od użytkowników nie można mieć pewności, czy oceniając określony element kierują się oni subiektywną oceną zgodności z pytaniem czy raczej swoimi bieżącymi potrzebami. Na gruncie RS miara wykorzystująca pojęcie relewancji zwana jest miarą dokładności klasyfikacji (ang. *classification accuracy*), gdyż mierzy częstość, z jaką system podejmuje poprawną lub niepoprawną decyzję co do faktu, czy element jest dobry czy też nie [Herlocker 2004].

W ramach niniejszego punktu zostaną przedstawione następujące miary efektywności: kompletność, dokładność, odrzut, kompletność porządkowa, mikrokompletność oraz mikrodokładność, tworzenie wykresu dokładności w funkcji kompletności, miara ROC (ang. *Relative Operating Characteristic*) oraz miary jednoliczbowe.

#### 4.4.1.1. Podstawowe miary efektywności IR

Kompletność  $R$  (ang. *recall* –  $R$ ), dokładność (ang. *precision* –  $P$ ), szum (ang. *fall-out* –  $F$ ) i ogólność (ang. *generality* –  $G$ ) wyszukiwania dla konkretnego pytania wyrażają się odpowiednio następującymi wzorami [Dąbrowski 1978]:

$$R = \frac{|A \cap B|}{|A|}, \quad (27)$$

$$P = \frac{|A \cap B|}{|B|}, \quad (28)$$

$$F = \frac{|\bar{A} \cap B|}{|\bar{A}|}, \quad (29)$$

$$G = \frac{|A|}{|A \cup \bar{A}|}, \quad (30)$$

gdzie:  $A$  – zbiór dokumentów relewantnych do danego pytania,

$B$  – zbiór dokumentów wyszukanych na dane pytanie,

$\bar{A}$  – dopełnienie zbioru  $A$ , czyli zbiór dokumentów nirelewantnych do danego pytania.

Kompletność określa zdolność systemu do wyszukiwania wszystkich relewantnych dokumentów, natomiast dokładność określa zdolność do wyszukiwania jedynie relewantnych dokumentów. Szum oznacza zdolność systemu do wyszukania dokumentów nirelewantnych, natomiast ogólność mierzy gęstość dokumentów relewantnych w całej kolekcji [Rijsbergen 1979]. Parametry te mogą mieć również inną interpretację na gruncie rachunku prawdopodobieństwa [Dąbrowski 1978]:  $R$  – prawdopodobieństwo, że dokument relewantny zostanie wyszukany,  $P$  – prawdopodobieństwo, że dokument wyszukany jest relewantny,  $F$  – prawdopodobieństwo, że dokument nirelewantny zostanie wyszukany,  $G$  – prawdopodobieństwo tego, że dokument jest relewantny.

Miary podstawowe wykorzystywane w przypadku, gdy zbiór odpowiedzi jest uporządkowany: kompletność porządkowa  $R_O$  i dokładność logarytmiczna  $P_{\log}$  [Salton 1975]. Wielkości te wyrażają się następującymi wzorami:

$$R_O = \frac{\sum_{i=1}^n i}{\sum_{i=1}^n r_i}, \quad (31)$$

$$P_{\log} = \frac{\sum_{i=1}^n \log i}{\sum_{i=1}^n \log r_i}, \quad (32)$$

gdzie:  $n$  – liczba dokumentów relewantnych,

$i$  – teoretyczna ranga dla pozycji relewantnej,

$r_i$  – rzeczywista ranga relewantnego dokumentu  $i$ .

Miary te są analogiczne do miar znormalizowanych [Salton 1975], są jednak od nich łatwiejsze do obliczenia. Podobnie jak miary znormalizowane są one funkcjami rang dokumentów relewantnych, chociaż nie zależą od liczności zbioru dokumentów  $N$ .

#### 4.4.1.2. Miary przeznaczone dla serii kwerend

Podstawą badania danego systemu nie jest skupianie się na pojedynczym pytaniu pojedynczego użytkownika, lecz analizowanie serii pytań często zadawanych i ocenionych przez wielu użytkowników. W tym wypadku istnieją dwa sposoby mierzenia kompletności i dokładności stosując mikro- i makroocenę [Salton 1975]. Makrooceny oparte są na zapytaniach kierowanych do systemu, tak więc każdy wskaźnik oparty na nich zależny jest od ich rozkładu (lub średniej). Jest to punkt widzenia użytkownika, który zadaje pytania i ocenia relewancję dokumentów pojawiających się w odpowiedzi. Średnia wartość każdego parametru (kompletności, dokładności itp.), która jest średnią ze wszystkich kwerend, stanowi ocenę wartości systemu dla przeciętnego użytkownika. Oczywiście, dla otrzymania wiarygodnych wyników wymaga się, aby ich rozkład był w miarę równomierny [Salton 1975]. Mikrooceny natomiast są oparte na własnościach dokumentów, czyli ogólnych własności relacji między zbiorami dokumentów relewantnych i nirelewantnych. Przy wykorzystaniu tego wskaźnika należy zapewnić, aby każda kwerenda dostarczała niezależnych danych, które spełniają odpowiednie kryteria relewancji. Należy również pamiętać, że kwerendy, które mają wiele elementów relewantnych mogą bardzo zniekształcić średnie wyniki. Dodatkowo należy zauważyć, że jeśli liczba dokumentów dla każdej kwerendy jest stała, to mikro- i makrooceny są sobie równe. Zostaną przedstawione parametry: mikrokompletność  $R_{\mu}$ , mikrodokładność  $P_{\mu}$ , makrokompletność  $R_m$  i makrodokładność  $P_m$ . Pozostałe parametry, czyli mikro- i makrosum, oraz ogólność można obliczyć analogicznie:

$$R_{\mu} = \frac{\sum_{i=1}^N |A_i \cap B_i|}{\sum_{i=1}^N |A_i|}, \quad (33)$$

$$P_{\mu} = \frac{\sum_{i=1}^N |A_i \cap B_i|}{\sum_{i=1}^N |B_i|}, \quad (34)$$

$$R_m = \frac{1}{N} \sum_{i=1}^N \frac{|A_i \cap B_i|}{|A_i|}, \quad (35)$$

$$P_m = \frac{1}{N} \sum_{i=1}^N \frac{|A_i \cap B_i|}{|B_i|}, \quad (36)$$

gdzie:  $N$  – całkowita liczba zadanych pytań przez różnych użytkowników,

$A_i$  – zbiór dokumentów relewantnych do pytania  $i$ ,

$B_i$  – zbiór dokumentów wyszukanych na pytanie  $i$ .

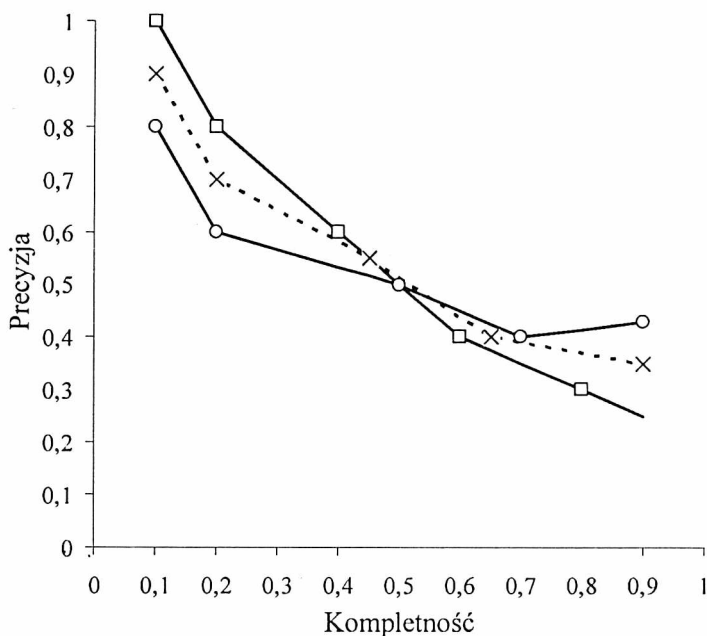
#### 4.4.1.3. Wykresy funkcji

Oprócz pojedynczych miar kompletności i dokładności do badania efektywności systemu w dziedzinie IR bardzo często używa się wykresu dokładności w funkcji kompletności. W literaturze można znaleźć przynajmniej kilka sposobów na wyznaczenie wykresu tej funkcji [Salton 1975], [Rijsbergen 1979]. Wykres dokładność–kompletność bardzo często służy do porównania różnych metod wyszukiwania, które zostały przeprowadzone na różnych zbiorach dokumentów, różnych kwerend i różnych użytkowników.

W pracy [Rijsbergen 1979] przedstawiono metodę tworzenia wykresu funkcji dokładność–kompletność (ang. *precision-recall* – P–R), znanej również jako krzywa dokładności–kompletności (ang. *precision-recall curve* – PRC), dla przypadku gdy odpowiedzi systemu wyszukiwania nie są uporządkowane. Różne punkty pary dokładność–kompletność uzyskuje się przez stopniowanie liczby wspólnych terminów, które występują w kwerendzie i dokumencie. Metoda ta może również zostać użyta w przypadku zastosowania miar mikro- i makro, które wykorzystują uśrednione wyniki – patrz rysunek 12 przedstawiający uśrednienie wyrażające mikrooceny. W praktyce może się zdarzyć, że określonej wartości kompletności nie odpowiada pojedyncza wartość dokładności, stąd konieczne jest zastosowanie odpowiednich technik interpolacji [Rijsbergen 1979].

Analizując typowe wykresy zależności P–R, można zauważyć, że dla niewielkich wartości kompletności (w okolicach zera), wartości dokładności są maksymalne (jeden lub poniżej jedności) oraz dla kompletności wynoszącej jeden, dokładność zwykle wynosi nieznacznie powyżej zera.

Inną metodę wyznaczania par P–R przedstawiono w pracy [Salton 1975], gdzie proponuje się wykorzystanie uporządkowania zbioru odpowiedzi. Należy tutaj dodać, że dla wielu metod wyszukiwania, np. wyszukiwania boolowskiego, takie uporządkowanie nie występuje, jednak w przypadku zastosowania modelu wektorowego i np. cosi-



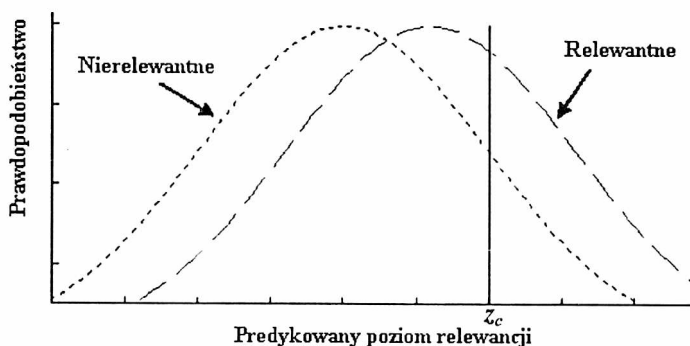
Rys. 12. Przykład uśredniania dla mikroocen [Rijsbergen 1979]

nusowej funkcji podobieństwa między reprezentacją kwerendy i dokumentu, w odpowiedzi na zapytanie uzyskiwany jest uporządkowany zbiór odpowiedzi. W takim wypadku można obliczać pary dokładność–kompletność dla wybranych kilku lub kilkunastu punktów określonych bądź to liczbą wyszukanych dokumentów relewanych lub ich procentem. Metodę tą można stosować zarówno dla pojedynczej kwerendy, jak i dla mikro- lub makrokompletności i dokładności.

Innego typu wykresem bazującym na relewanościach dokumentów, jednak alternatywnym do par P–R jest opracowana przez Swetsa [Rijsbergen 1979] miara krzywoliniowa ROC znana pod nazwą relatywnej charakterystyki operacyjnej (ang. *relative operating characteristic* – ROC) lub operacyjnej charakterystyki odbioru (ang. *receiver operating characteristic* – ROC), która to nazwa wywodzi się z wypracowania miary ROC na gruncie teorii wykrywania sygnałów (ang. *signal detection theory*) [Herlocker 2004]. Bez względu na to, skąd się wywodzi i gdzie jest wykorzystywana, miara ta służy do rozróżnienia sygnału (relewancji) od szumu (nierelewancji). Zakłada się, że każdemu dokumentowi system informacyjny przyporządkowuje przewidywaną wartość relewancji. Możliwą dystrybucję prawdopodobieństw dokładności względem relewancji i nierelewancji dokumentów przedstawia rysunek 13. Pola pod wykresami obiektów relewanych i nierelewanych można traktować odpowiednio jako: kompletność (R) i szum (F). Interpretacja tych wykresów jest oczywista: im bardziej rozsunięte są te wykresy, tym jakość predykcji systemu jest lepsza.

Dla wielu modeli zakłada się istnienie pewnej wartości poziomu relewancji  $z_c$ , na którym następuje odcięcie na elementy wyszukane i niewyszukane. Poziom relewancji



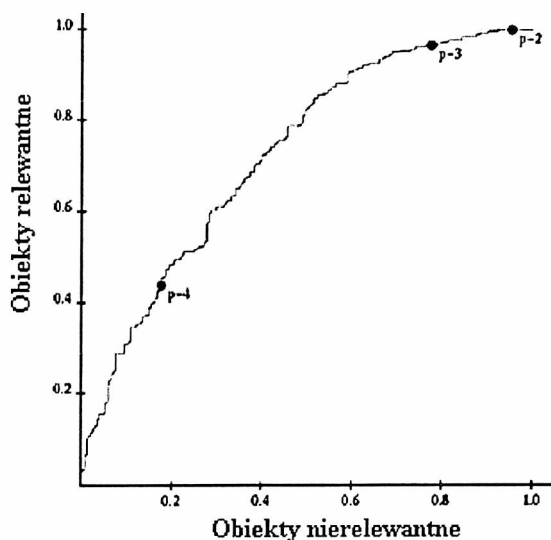


Rys. 13. Możliwy rozkład prawdopodobieństwa relevantnych i nirelevantnych obiektów [Herlocker 2004]

może być utożsamiany z pozycją na posortowanej liście odpowiedzi występującej w wielu systemach ocenie określonego obiektu.

Dla każdego punktu na osi rzędnych, znajdującego się na prawo od założonego poziomu relewancji  $z_c$  można wyznaczyć dwie wartości kompletności ( $R$ ) i szumu ( $F$ ). Na podstawie tych par można wyznaczyć kolejny wykres zwany ROC będący zależnością kompletności od szumu lub inaczej zależnością procentu obiektów relevantnych od procentu obiektów nirelevantnych. W pracy [Herlocker 2004] można znaleźć prostą metodę tworzenia tego wykresu (patrz rys. 14). Przedstawia się ona następująco:

1. Wyznacz metodę identyfikacji elementu jako relevantnego lub nirelevantnego (np. dla bazy *MovieLens* może to być rzeczywista ocena filmu 4 lub 5).
2. Wygeneruj ranking elementów (dla bazy *MovieLens* predykcje ocen pewnych filmów).



Rys. 14. Wykres ROC [Herlocker 2004]

3. Każdy element, dla którego mamy predykcję, uporządkuj według malejącej wartości predykcji relewancji, a następnie przystap w kolejności do ich analizy rysując graf od punktu (0,0):

- a) jeżeli element jest relewantny narysuj jeden krok w pionie,
- b) jeżeli element jest nierелеwantny narysuj jeden krok w poziomie,
- c) jeżeli nie posiadamy informacji na temat relewancji elementu ignorujemy go.

Na wykresie (rys. 5) można również zaznaczyć punkty poziomu relewancji, które zostały przedstawione jako  $p = 4$ ,  $p = 3$  i  $p = 2$ , oznaczające wartość predykcji określonego elementu (tutaj oceny filmu z bazy *MovieLens*).

#### 4.4.1.4. Miary jednoliczbowe

W dziedzinie IR opracowano również miary jednoliczbowe, które w sposób syntetyczny wyrażają efektywność systemów wyszukiwania informacji. Główną przyczyną ich powstania było niezadowolenie z używania miar złożonych [Rijsbergen 1979]. Dalej zostaną przedstawione takie syntetyczne miary, jak: suma  $S$ , miara opracowana przez Borko  $BK$  i miara  $Q$  wykorzystująca oprócz wielkości  $P$  i  $R$ , wielkość szumu  $F$ :

$$S = P + R, \quad (37)$$

$$BK = P + R - 1, \quad (38)$$

$$Q = \frac{R - F}{R + F - 2RF}. \quad (39)$$

#### 4.4.2. Pozostałe miary badania efektywności systemów rekomendacyjnych

Oprócz miar wywodzących się z dziedziny IR w dziedzinie RS wykorzystuje się również miary powszechnie stosowane w badaniu efektywności systemów różnego typu [Herlocker 2004]. Należą do nich: średni błąd absolutny (MAE) oraz pochodne do niego: średni błąd kwadratowy (MSE), znormalizowany MAE (NMAE), pierwiastek ze średniego błędu kwadratowego (RMSE), standardowe odchylenie błędów ( $\sigma_F$ ), średni błąd absolutny (MUAE) i miara *All but one* MAE (ABOMAE). Oprócz miar bazujących na MAE w dziedzinie RS często wykorzystuje się trzy inne miary: miarę przydatności (ang. *utility*), *half-life* oraz znormalizowaną miarę opartą na funkcji odległości (ang. *normalized distance-based performance measure* – NDPM). Miary te, w odróżnieniu od miar opartych na pojęciu relewancji (lub inaczej mierzących dokładność klasyfikacji), bazują na pewnej skali ocen, tak jak w przypadku systemu *MovieLens* [GroupLens 2008].

##### 4.4.2.1. MAE i pochodne

Średni błąd absolutny (ang. *mean absolute error* – MAE) jest jedną z najczęściej używanych metryk w dziedzinie RS: mierzy on średnie absolutne odchylenie między predykcją oceny a jej rzeczywistą wartością. Pierwsze zastosowania tej metryki w dziedzinie systemów rekomendacyjnych można odnaleźć we wczesnych pracach nad filtrowaniem kolaboratywnym [Shardanand 1995], [Breese 1998] i [Herlocker 1999]. MAE wyraża się następującym wzorem:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |p_i - r_i|, \quad (40)$$

gdzie:  $N$  – liczba predykcji,

$p_i$  –  $i$ -ta predykcja oceny,

$r_i$  –  $i$ -ta rzeczywista ocena.

Miara MAE wykazuje małe błędy tych systemów, które generują odpowiednio dobre predykcje dla całego zbioru rekomendowanych elementów [McLaughlin 2004]. MAE oprócz swoich zalet w mierzeniu efektywności systemów rekomendacyjnych z predykcją ocen ma jednak pewne wady. Należą do nich [McLaughlin 2004] i [Herlocker 2004]:

- brak rzeczywistych ocen dla określonych predykcji,
- mniejsza przydatność gdy użytkownik ogląda jedynie pierwsze rekomendowane elementy, gdyż w takim przypadku mały błąd predykcji elementów, których użytkownik nigdy nie zobaczy, jest zupełnie nieistotny,
- nie daje się zastosować w przypadku algorytmów z predykcją rang, ale bez predykcji ocen,
- zależność od stosowanej skali ocen, stąd nie nadaje się do porównywania wyników działania systemów wykorzystujących różne skale ocen.

W celu uzyskania porównywalnych wyników dla różnej skali ocen można zastosować znormalizowany średni błąd absolutny (ang. *normalised mean absolute error* – NMAE):

$$\text{NMAE} = \frac{\text{MAE}}{r_{\max} - r_{\min}}, \quad (41)$$

gdzie:  $r_{\max}$  – maksymalna ocena,

$r_{\min}$  – minimalna ocena.

W niektórych pracach [Chen 2004], [Sobecki 2007] wykorzystuje się miarę zwaną dokładnością predykcji (ang. *prediction accuracy* – PA), która obliczana jest jako NMAE odjęte od jedności:

$$\text{PA} = 1 - \text{NMAE}. \quad (42)$$

Miara PA jest w pewnym sensie analogiczna do miary dokładności P, tzn. idealna jej wartość wynosi 1, a najgorsza 0.

Kolejnymi miarami bazującymi na MAE są: średni błąd kwadratowy (ang. *mean squared error* – MSE) i pierwiastek ze średniego błędu kwadratowego (ang. *root mean squared error* – RMSE), które bardziej uwidaczniają występujące błędy predykcji [Herlocker 2004]. Miary te, niestety, dziedziczą wady MAE, są jednak od niej o wiele rzadziej wykorzystywane:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (p_i - r_i)^2, \quad (43)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (p_i - r_i)^2}. \quad (44)$$

Na bazie MAE można również skonstruować miarę średniego odchylenia standardowego błędów, która pokazuje na ile spójna jest predykcja, czyli im mniejsze wartości odchylenia standardowego błędu, tym spójność jest większa [Shardanand 1995]. Odchylenie standardowe błędu  $\sigma_E$  wyraża się następującym wzorem:

$$\sigma_E = \sqrt{\frac{\sum_{i=1}^N (E_i - \text{MAE})^2}{N}}, \quad (45)$$

gdzie:  $E_i$  – oznacza błąd  $i$ -tej predykcji,  $E_i = p_i - r_i$ .

W pracy [Massa 2004] można znaleźć następną modyfikację miary MAE; odnosi się ona do poszczególnych użytkowników i nosi nazwę średni absolutny błąd użytkownika (ang. *mean user absolute error* – MUAE), który obliczany jest jako średnia błędów MAE dla poszczególnych użytkowników. Miara ta, pod inną nazwą, została wykorzystana w pracy [Lemire 2005], gdzie nadano jej nazwę All but One MAE (ABOMAE). W ten sposób błąd dla każdego użytkownika brany jest tylko raz, co powoduje, że użytkownicy, którzy ocenili bardzo wiele elementów nie mają tak wielkiego wpływu na całą miarę błędu, a to eliminuje jedną z wad miary MAE. MUAE wyraża się następującym wzorem:

$$\text{MUAE} = \frac{1}{K} \sum_{i=1}^K \text{MAE}_i, \quad (46)$$

gdzie:  $K$  – liczba użytkowników,

$\text{MAE}_i$  – MAE dla poszczególnego użytkownika  $i$ .

## Pozostałe miary efektywności RS

W ramach badania RS opracowano również inne miary i metody badania efektywności, które nie są oparte na MAE; są to miary jakości predykowanej oceny: miara przydatności (ang. *utility half-life*) oraz miara NDPM, oraz miary, które odnoszą się do jakości niezwiązanych z predykowanymi ocenami, czyli związane z pokryciem rekomendacji.

**Miara przydatności** została opracowana przez Breese'a i innych [Breese 1998] i służy ona do określenia, jak bardzo prawdopodobne jest przeglądanie przez użytkownika poszczególnego elementu na liście uporządkowanej odpowiedzi. W mierze tej zakłada się, że prawdopodobieństwo to maleje eksponentialnie. Miara ta przeznaczona jest do mierzenia przydatności dla użytkownika rekomendowanego obiektu, gdzie przydatnością nazywa się różnicę między rekomendowaną oceną, a oceną neutralną, która jest

charakterystyczną dla dziedziny właściwej dla rekomendowanych obiektów. Miara przydatności  $Ru_i$  dla użytkownika  $i$ , wyraża się następującym wzorem:

$$Ru_i = \sum_{j=1}^n \frac{\max(r_{i,j} - d, 0)}{\frac{j-1}{2^{\alpha-1}}}, \quad (47)$$

gdzie:  $j$  – pozycja obiektu na liście uporządkowanej (według malejącej wartości predykcji oceny liście rekomendowanych obiektów),

$n$  – liczba elementów na rekomendowanej liście,

$d$  – ocena neutralna, która jest charakterystyczna dla dziedziny właściwej dla rekomendowanych obiektów,

$\alpha$  – numer pozycji elementu, którego prawdopodobieństwo obejrzenia wynosi 0,5,

$r_{i,j}$  – ocena obiektu na pozycji  $j$  przyznana przez użytkownika  $i$ .

Wzór (47) wykorzystywany jest do wyznaczenia przydatności dla poszczególnego użytkownika; w przypadku gdy dla jakiegoś elementu z listy brak jest oceny użytkownika wówczas nadawna jest mu wartość  $d$ , co praktycznie eliminuje go z obliczeń.

Całkowita miara dla całego zbioru użytkowników  $Ru$  obliczana jest według następującego wzoru:

$$Ru = 100 \frac{\sum_{i=1}^N Ru_i}{\sum_{i=1}^N Ru_i^{\max}}, \quad (48)$$

gdzie:  $N$  – liczba analizowanych użytkowników,

$Ru_i^{\max}$  – maksymalna przydatność, którą można byłoby osiągnąć, jeśli wszystkie elementy byłyby na czele listy.

Miara przydatności umożliwia analizowanie efektywności systemów bez względu na rozmiar zbioru testowego oraz liczbę rekomendowanych elementów. Miara ta jest niewrażliwa na niewłaściwie przyznanie bardzo niskiej oceny, ponieważ traktuje ona w ten sam sposób wszystkie oceny niższe od neutralnej. W pracy [Herlocker 2004] wskazuje się dodatkowo na możliwy brak uniwersalności tej metody z powodu doboru różnych wartości parametru  $\alpha$  w różnych badaniach, co spowodowane jest koniecznością jego dostosowania do zróżnicowania użytkowników różnych systemów.

**Znormalizowana miara oparta na funkcji odległości** (ang. *normalized distance-based performance measure* – NDPM) została opracowana przez Yao w 1995 roku i zastosowana w systemie rekomendacyjnym *FAB* [Herlocker 2004]. Miara NDPM może zostać wykorzystana do porównania dwóch słabouporzędkowanych rankingów. Opiera się ona na podejściu wypracowanym na gruncie teorii decyzji i pomiaru (ang. *theory of decision and measurement*). Miara NDPM określona jest następującym wzorem:

$$\text{NDPM} = \frac{2C^- + C''}{2C^i}, \quad (49)$$

gdzie:  $C^-$  – liczba sprzeczności występujących między uporządkowaniem rzeczywistych ocen i ich predykcji,

$C''$  – liczba niesprzeczności występujących między uporządkowaniem rzeczywistych ocen i ich predykcji,

$C^i$  – liczba wszystkich par elementów.

Miara NDPM jest znormalizowana, więc może być wykorzystana dla różnych zbiorów danych; licznik reprezentuje tu odległość, a mianownik największą możliwą odległość [Herlocker 2004]. Miara NDPM jest w pewnym sensie analogiczna do miary podobieństwa  $\tau$  Kendalla zdefiniowanej wzorem (14), dostarcza jednak lepszej interpretacji dla niesprzeczności, przy czym jest jednakowo czuła na sprzeczności występujące zarówno na początku, jak i na końcu listy rekomendowanych elementów. Ponieważ miara ta nie „penalizuje” sytuacji, gdy występuje niesprzeczność w porządku, lepiej nadaje się ona do takich zastosowań, w których można poprzestać na „wystarczająco dobrych” rezultatach.

**Miara pokrycia** (ang. *coverage*) obliczana jest jako stosunek liczby elementów dla, których algorytm jest zdolny do dostarczenia rekomendacji, do wszystkich elementów w bazie [Papagelis 2005]. Miara pokrycia zbioru rekomendacji wyraża się wzorem:

$$\text{Coverage} = \frac{|Pd|}{N}, \quad (50)$$

gdzie:  $Pd$  – zbiór, dla którego RS jest w stanie dostarczyć predykcji,

$N$  – liczba elementów w całym zbiorze.

W wielu metodach rekomendacji, np. filtrowaniu kolaboratywnym (CF), narzucanie warunków na minimalną wielkość korelacji lub liczby wspólnych dokumentów, powoduje, że pokrycie maleje. W literaturze [Papagelis 2005] można znaleźć badania efektywności różnych algorytmów rekomendacji wyrażonych miarą MAE, na różnych poziomach rzadkości (ang. *sparsity*) rekomendacji.

#### 4.4.3. Wykorzystanie miar efektywności do badania rekomendacji w zbiorze *MovieLens*

W niniejszym punkcie zostaną przedstawione przykładowe wyniki badań nad efektywnością rekomendacji z wykorzystaniem zbioru *MovieLens* z 1 milionem ocen. Parametrami, które badano dla różnych funkcji podobieństwa, były przedstawione powyżej miary efektywności: MAE, NMAE, MSE, dokładność predykcji oraz pokrycie. Dla porównania przedstawiono również czas realizacji wszystkich obliczeń i średnio na jednego użytkownika. Badania, tak jak w przypadku wyznaczania macierzy podobieństw zaprezentowanego w punkcie 4.3.1, zostały przeprowadzone w zbliżonych warunkach uruchomieniowych na komputerze o następującej konfiguracji: procesor 3 GHz Dual

Tabela 4. Wyniki badania efektywności predykcji rekomendacji dla zbioru 1 milion ocen *MovieLens*

| Parametry metody                                                                                                                            | Czas<br>s | Średni czas<br>dla jednego<br>użytkownika<br>s | MAE     | NMAE    | MSE     | Dokładność<br>predykcji | Pokrycie |
|---------------------------------------------------------------------------------------------------------------------------------------------|-----------|------------------------------------------------|---------|---------|---------|-------------------------|----------|
| Podobieństwo Pearsona, min. 50 wspólnych ocen, sąsiedztwo 30 użytkowników, predykcja średnia ocen                                           | 87469,08  | 14,48                                          | 0,72748 | 0,18187 | 0,82796 | 81,81%                  | 67,66%   |
| Podobieństwo cosinusowe, min. 50 wspólnych ocen, sąsiedztwo 30 użytkowników, predykcja średnia ocen                                         | 95128,57  | 15,75                                          | 0,75201 | 0,18800 | 0,90774 | 81,20%                  | 67,81%   |
| Podobieństwo współczynnika ważonego $\kappa$ Chena liniowego, min. 50 wspólnych ocen, sąsiedztwo 30 użytkowników, predykcja średnia ocen    | 85449,82  | 14,15                                          | 0,69419 | 0,17355 | 0,78426 | 82,65%                  | 67,71%   |
| Podobieństwo współczynnika ważonego $\kappa$ Chena kwadratowego, min. 50 wspólnych ocen, sąsiedztwo 30 użytkowników, predykcja średnia ocen | 86080,04  | 14,25                                          | 0,69796 | 0,17449 | 0,77255 | 82,55%                  | 67,66%   |
| Podobieństwo $\tau$ Kendalla, min. 50 wspólnych ocen, sąsiedztwo 30 użytkowników, predykcja średnia ocen                                    | 86452,81  | 14,31                                          | 0,73765 | 0,18441 | 0,85139 | 81,56%                  | 67,71%   |



Core, RAM 3 GB, system MS Windows XP Professional. Obliczona macierz podobieństwa każdą z funkcji podobieństwa stanowiła parametr wejściowy do procedury wyznaczania predykcji. W tabeli 4 podano wyniki badania predykcji rekomendacji dla wybranych metod.

Na podstawie wyników przeprowadzonych badań można zauważyć, że dokładność predykcji jest porównywalna dla większości funkcji podobieństwa, przy czym największa jest dla funkcji ważonego współczynnika  $\kappa$  Chena w wersji liniowej. Czas obliczeń nawet dla pojedynczego użytkownika jest dość spory, przy czym średnio w tym czasie była określana Predykcja dla około 166 filmów. W badanych metodach pokrycie praktycznie jest równe we wszystkich badanych przypadkach. Warto zwrócić uwagę, że przy wykorzystaniu miary MSE wyniki są bardziej zróżnicowane niż w przypadku innych miar wprost pochodnych od MAE (NMAE i dokładność predykcji).

## 4.5. Podsumowanie

Przedstawiono różne metody wykorzystania profilu użytkownika do wyznaczenia rekomendacji. Omówiono stosowane w systemach rekomendacyjnych metody filtrowania informacji: DF, CBF, CF, IBCF, RBCF oraz HA. Wkład pracy autora w niniejszej dziedzinie polegał z jednej strony na zastosowaniu klasycznych metod filtrowania do rekomendacji interfejsów, np. DF w systemie *Megane* [Sobecki 2004b] i CF w systemie *Reporter* [Sobecki 2007], na zastosowaniu wnioskowania rozmytego do DF w systemie *Kucharek* [Sobecki 2006] oraz bazującej na consensusie metodzie HA [Sobecki 2005].

Przedstawiono techniki dopasowania rekomendowanego elementu do profilu użytkownika. Większość z powszechnie stosowanych metod wykorzystuje funkcje podobieństwa lub odległości między profilami. W niniejszym punkcie przedstawiono definicje i własności dotyczących przekształcania funkcji odległości i podobieństwa, następnie zostały zaprezentowane najpopularniejsze funkcje stosowane w RS oraz porównanie czasu obliczeń macierzy podobieństwa przez wybrane funkcje podobieństwa w zbiorze *MovieLens*.

Przedstawiono problemy dopasowania między profilami użytkowników, przez znajdowanie podobnych użytkowników, realizowane przez zastosowanie odpowiednich funkcji podobieństwa. Mając określone podobieństwa w przestrzeni użytkowników, można wyznaczyć sąsiedztwo określonego użytkownika, czyli grupę najbardziej podobnych użytkowników na podstawie wartości funkcji podobieństwa oraz innych warunków, np. liczby wspólnie ocenionych elementów. Natomiast po określeniu sąsiedztwa określonego użytkownika można wyznaczyć predykcję oceny elementu na podstawie ocen dostarczonych przez użytkowników z sąsiedztwa. Przedstawiono metody wyznaczania predykcji, od najprostszych uwzględniających średnią ocen po bardziej złożone metody uwzględniające normalizację wyrażoną odchyleniami standardowymi dla wartości ocen użytkownika, dla którego wyznaczamy predykcję, i poszczególnych użytkowników z jego sąsiedztwa.

Zaprezentowano metody oceny efektywności RS. Metody te zostały podzielone m.in. ze względu na źródło ich pochodzenia, tzn. metody opracowane na gruncie IR, oraz pozostałe metody ze szczególnym uwzględnieniem metod opartych na MAE. Przedstawiono również weryfikację eksperymentalną wybranych metod wyznaczania sąsiedztwa dla zbioru 1 milion ocen *MovieLens*, przy metodzie wyznaczania predykcji za pomocą średniej ocen, badając czasy obliczeń wyznaczania predykcji, MAE, NMAE, PA oraz pokrycie. Najlepsze rezultaty dokładności predykcji (PA) otrzymano dla współczynnika ważonego  $\kappa$  Chena liniowego, przy równie dużym współczynniku pokrycia.



## 5. Adaptacja interfejsów użytkownika metodą consensusu

Współczesne systemy webowe powinny mieć interfejs użytkownika przystosowany do wykorzystania przez coraz bardziej zróżnicowanych użytkowników. Jednym ze sposobów umożliwiających dostosowanie interfejsu do potrzeb i gustów użytkowników jest personalizacja [Mobasher 2000]. Definicja mówi, że personalizowane hipermedia są systemami, które adaptują swoją zawartość, strukturę i/lub sposób prezentacji obiektów do charakterystyki, zachowań i sposobów wykorzystania tych systemów przez użytkownika [Kobsa 2001]. Termin personalizacja w literaturze przedmiotu jest więc pojmowany podobnie jak adaptacja systemów webowych.

Personalizacja, zarówno manualna jak i automatyczna stała się obecnie nieodłącznym składnikiem wielu webowych serwisów informacyjnych, a zwłaszcza portali informacyjnych, takich jak Lycos, Yahoo, ale także polskiej Wirtualnej Polski i nowej wersji portalu Onet.

Zasadniczo, bardzo podobnymi do problemów personalizacji zajmują się specjaliści z dziedziny inteligentnych agentów (ang. *Intelligent Agent*) [Wooldridge, 1995]. Pomiędzy fachowcy z tej dziedziny wywodzą się z odmiennych kręgów niż fachowcy z dziedziny HCI [Prece 1996], bardzo często zarówno przedmioty ich zainteresowań, jak i proponowane rozwiązania są bardzo podobne. Pewne trudności w analizie prac powstających w ramach tych dziedzin może stanowić odmienna terminologia.

W niniejszym rozdziale przedstawiono zastosowanie metod consensusu do rekomendacji interfejsów użytkownika na przykładzie webowego systemu informacyjnego o wybranym typie samochodu – *Megane* [Sobecki 2004b]. Korzystając z typologii systemów rekomendacyjnych zaprezentowanej w rozdziałach 3 i 4 oraz zdefiniowanych tam pojęć można w skrócie scharakteryzować poszczególne elementy systemu z adaptacją interfejsu wykorzystującą metody consensusu, który zostanie przedstawiony w drugim punkcie niniejszego rozdziału. Dziedziną zastosowania prezentowanego systemu rekomendacyjnego jest adaptacja interfejsów użytkownika. Jednym z zadań niektórych użytkowników tego systemu jest sprofilowanie interfejsu, jednak podstawowym jest zapoznanie się z prezentowanymi informacjami. Zawartość profilu użytkownika obejmuje zarówno dane użytkownika w postaci pewnych danych demograficznych, jak i dane użytkownika w postaci konfiguracji interfejsu użytkownika wraz z ich oceną dokonaną przez użytkownika. Do reprezentacji profilu została wykorzystana krotka. Profil inicjowany jest manualnie. W systemie nie stosuje się żadnych technik uczenia profilu, gdyż dane dotyczące konfiguracji interfejsu pozyskiwane są w sposób bezpośredni. W syste-

mie korzysta się ze sprzężenia zwrotnego polegającego na ocenie satysfakcji z konkretnego interfejsu użytkownika. Profil może być adaptowany manualnie.

Wykorzystanie profilu odbywa się za pomocą podejścia hybrydowego (HA) przez filtrowanie demograficzne (DF) i filtrowanie kolaboratywne (CF). Dopasowanie rekomendowanego interfejsu użytkownika do profilu użytkownika odbywa się na podstawie klasyfikacji. W ramach dopasowania między profilami, znajdowanie podobnych użytkowników realizowane jest przez zastosowanie własnej funkcji odległości, natomiast utworzenie sąsiedztwa jest realizowane na podstawie grupowania. Wyznaczenie predykcji wybranego sąsiedztwa odbywa się na podstawie zastosowania metod consensusu. Przedstawiono tam zarówno elementy teorii consensusu w zastosowaniu do rekomendacji interfejsów [Nguyen 2003a] oraz jej modyfikacje obejmujące przystosowanie do wykorzystania przez różne metody filtrowania informacji DF, CBF i CF [Sobecki 2005].

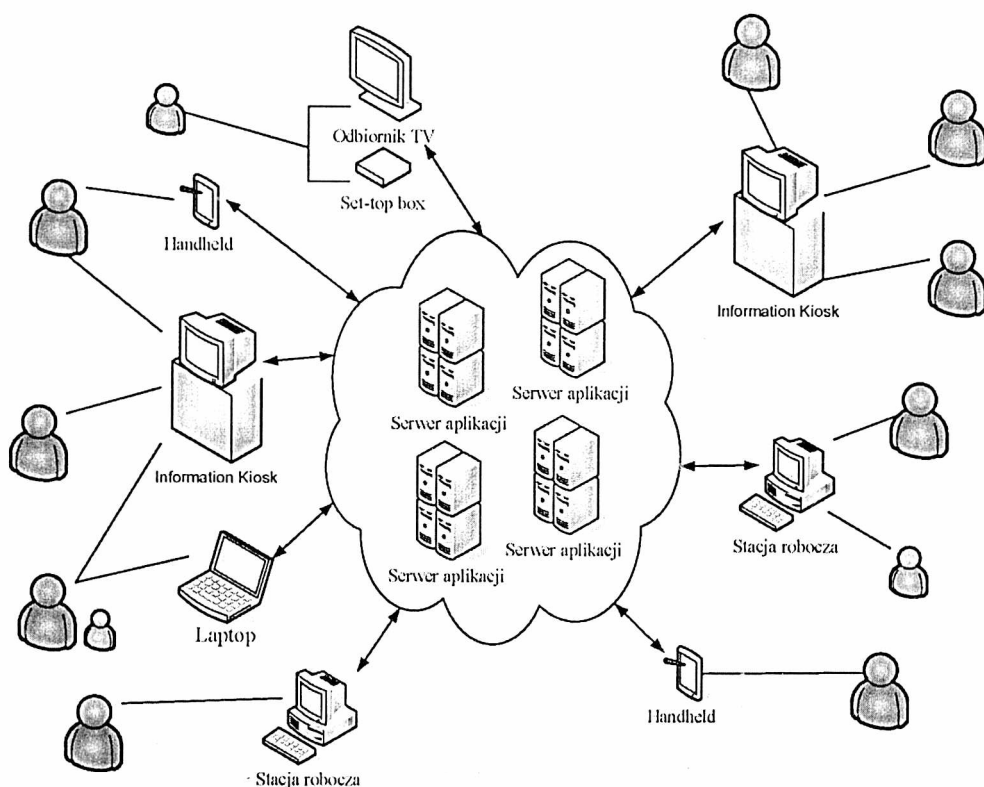
Przedstawiono architekturę systemu *Megane* [Sobecki 2004b] oraz teorię consensusu. Omówiono problemy reprezentacji, tworzenia i utrzymania profilu użytkownika oraz wykorzystania profilu. Opisano implementację systemu *Megane* oraz badanie eksperymentalne dotyczące użyteczności rekomendowanego interfejsu systemu *Megane*.

Indywidualny wkład Autora w rozwój koncepcji zastosowania metod consensusu do adaptacji interfejsów użytkownika w systemach webowych obejmuje: określenie tego problemu, zaproponowanie architektury adaptacji interfejsu użytkownika, opracowanie modelu użytkownika obejmującego profile użytkownika i profile interfejsu użytkownika oraz metod reprezentacji, inicjowania, modyfikacji i uczenia profilu użytkownika, opracowanie hybrydowego podejścia HA do rekomendacji interfejsu użytkownika wykorzystującego metody consensusu, opracowanie założeń projektowych do systemu *Megane* służącego do eksperymentalnej weryfikacji zaproponowanych metod, przeprowadzenie eksperymentu oraz opracowanie wniosków na podstawie osiągniętych wyników.

## 5.1. Architektura adaptacyjnego interfejsu użytkownika webowego systemu informacyjnego

System *Megane* należy do klasy systemów wieloagentkich (ang. *multiagent system*), gdzie podstawowym zadaniem każdego agenta jest dostarczanie użytecznych interfejsów użytkownika [ISO 1997]. Zróżnicowanie środowiska, w którym funkcjonuje system przedstawiono na rysunku 15. Schemat ten przedstawia społeczność zróżnicowanych i rozproszonych użytkowników wykorzystujących zróżnicowane platformy systemowo-sprzętowe celem uzyskania dostępu do określonego webowego systemu informacyjnego.

Implementacja tego systemu stosuje architekturę trójwarstwową, dlatego zasoby i funkcje systemu są przynajmniej częściowo rozproszone. Użytkownicy logują się do systemu przez platformę klienta, którego zadaniem jest praktyczna realizacja całego interfejsu użytkownika. Użytkownicy rejestrują się w systemie, wypełniając formularz.



Rys. 15. Zróżnicowanie środowiska systemu adaptacyjnego

Dane pobrane z tego formularza tworzą (przynajmniej częściowy) profil użytkownika. Profil użytkownika zapisywany jest przez warstwę aplikacji w warstwie bazy danych.

Na podstawie danych zawartych w profilu użytkownika i zapisanych w bazie danych, warstwa aplikacji generuje konkretną instancję interfejsu dla użytkownika, której postać zapisywana jest w tej części profilu użytkownika, która zawiera jego dane. Ze względu na możliwość występowania różnych wersji zawartości, struktury i wyglądu interfejsu, baza danych jest również wykorzystywana do ich przechowywania, a warstwa aplikacji musi pobrać określone dane z bazy przed ich przetworzeniem i przesłaniem do warstwy klienta.

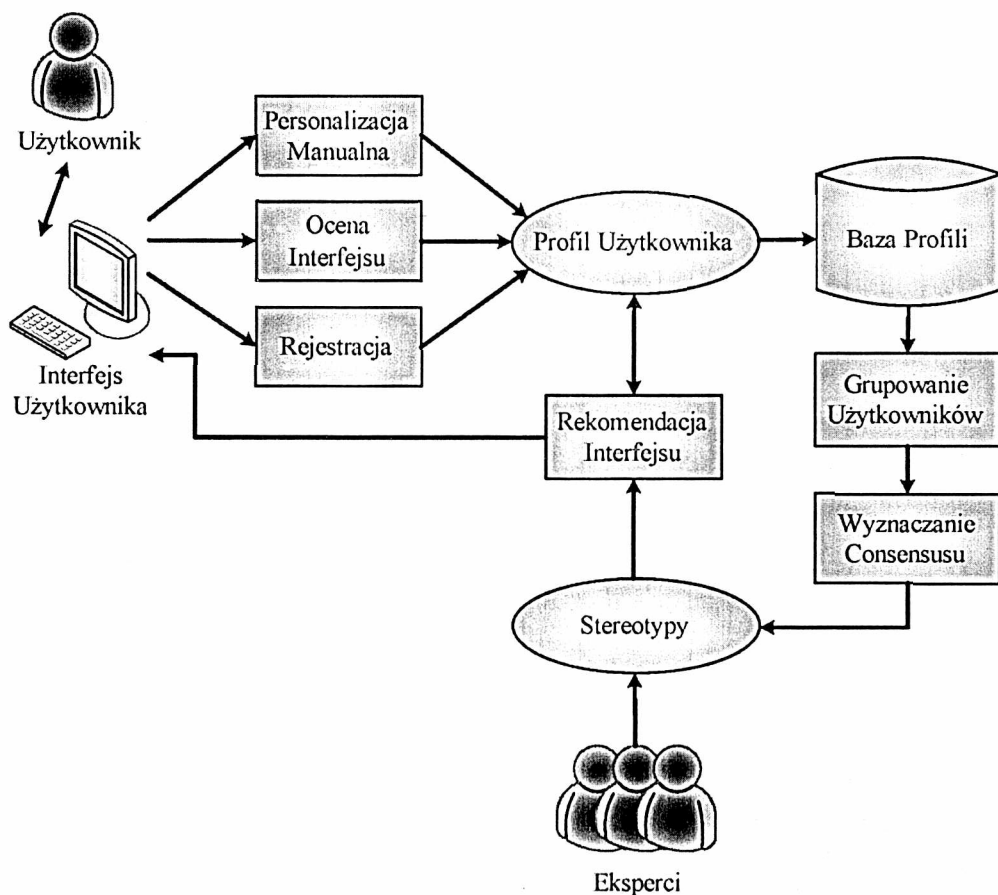
Zarejestrowany i zalogowany użytkownik ma możliwość manualnej personalizacji interfejsu, zmieniając zarówno jego wygląd, jak i zawartość, np. szczegółowość prezentowanych informacji. Spersonalizowany interfejs przechowywany jest w tej części profilu użytkownika, która obejmuje instancję interfejsu użytkownika. Użytkownik ma również możliwość oceny użyteczności tego interfejsu, która to ocena również jest przechowywana w profilu.

Zaprezentowany fragment architektury adaptacyjnych interfejsów użytkownika webowego systemu informacyjnego bazuje wyłącznie na manualnym dostosowaniu inter-

fejsów dla konkretnych użytkowników, które nie podlegają automatycznej zmianie bez jego wyraźnej zgody. Jednak ten etap jest niezbędny do realizacji kolaboratywnej adaptacji opartej na opiniach wielu użytkowników systemu. Kompletną architekturę systemu prezentuje rysunek 16.

Praca systemu podzielona jest więc na dwa etapy. W pierwszym zbierane są dane potrzebne do adaptacji, tzn. użytkownicy rejestrują się w systemie, logują się, wyznaczany jest dla nich interfejs użytkownika na podstawie wnioskowania stereotypowego, i w zależności od swoich potrzeb personalizują interfejs, który następnie mogą ocenić. W drugim etapie, na podstawie danych zebranych w pierwszym etapie modyfikowane są stereotypy z wykorzystaniem grupowania użytkowników i zastosowania metod consensusu do wyznaczenia nowych konfiguracji interfejsu dla określonych grup.

W celu lepszego dostosowania interfejsu dla różnych użytkowników, projektanci systemu (eksperci) wyznaczyli kilka stereotypów, które przyporządkowują centroidom wartości demograficznych określone konfiguracje interfejsów użytkownika. Każdy nowy użytkownik, po zarejestrowaniu, przydzielany jest do określonej klasy, przez znalezie-



Rys. 16. Architektura adaptacyjnego interfejsu użytkownika



nie centroidu, do którego dane użytkownika zawarte w jego profilu mają najmniejszą odległość (w przypadku gdy odległości do wszystkich centroidów są większe od określonego progu, użytkownik przyporządkowany zostaje do klasy użytkowników „domyślnych”).

Przejsie systemu do drugiego etapu uwarunkowane jest zarejestrowaniem wystarczającej liczby użytkowników. W etapie tym, na podstawie zgromadzonych profili użytkownika, następuje ich pogrupowanie za pomocą algorytmu Dattoli lub  $k$ -średnich (algorytmy opisano w punkcie 4.2.3). W rezultacie działania algorytmu powstają grupy użytkowników oraz centroidy tych grup, które są zapamiętywane do wykorzystania w dalszych etapach pracy systemu. Następnie dla każdej grupy użytkowników wybierane są takie konfiguracje interfejsów, które uzyskały wystarczająco pozytywne oceny użyteczności. Z tych profili interfejsów wyznaczany jest consensus. W przypadku gdy interfejs reprezentowany jest przez krotkę, consensusem jest consensus dla każdej wartości atrybutu. Consensusy te wyznaczane są na podstawie odpowiednich algorytmów dostosowanych do mikrostruktury tych wartości [Nguyen 2003a]. W efekcie działania wyboru consensusu dla każdej z grup wyznaczona zostaje rekomendacja dla danej grupy reprezentowanej przez centroid profilu użytkownika danej grupy.

Po zrealizowaniu podanych czynności każdy nowy użytkownik klasyfikowany jest na podstawie nowych centroidów oraz przydzielane mu są nowe konfiguracje interfejsu wyznaczone na drodze consensusu. Czynności grupowania i wyznaczania nowych consensusów można powtarzać począwszy od momentu, w którym system zgromadzi odpowiednią liczbę nowych użytkowników lub gdy oceny użyteczności interfejsów rekomendowanych przez system adaptacyjny drastycznie się obniżą.

## 5.2. Teoria consensusu

Teoria consensusu (ang. *Consensus Theory* – CT) powstała w ramach nauk socjologicznych, wywodzi swoje korzenie z teorii wyboru [Aizerman 1985], gdzie jej podstawowy problem zdefiniowano następująco: *Niech będzie dany zbiór  $X$  (alternatyw, obiektów) będący podzbiorem uniwersum  $U$ . Wybór polega na selekcji na podstawie określonych kryteriów pewnego podzbioru zbioru  $X$ .*

Teoria consensusu CT nie narzuca, aby rezultat wyboru należał do zbioru  $X$  ani nawet, aby miał taką samą strukturę, co jest konieczne w przypadku teorii wyboru. Teoria consensusu, podobnie jak eksploracja danych (ang. *data mining*), zajmuje się problemami analizy danych w celu wydobywania użytecznych informacji, natomiast główna różnica między nimi polega na tym, że metody eksploracji danych specjalizują się w odkrywaniu wzorców w danych, gdyż za ich pomocą można poszukiwać związków „przyczynowo-skutkowych”, które są ukryte w danych, natomiast metody consensusu umożliwiają wyznaczenie takiej instancji dla pewnego zestawu wersji danych, która najlepiej je reprezentuje lub jest dobrym kompromisem, który powinien być akceptowany przez strony w konflikcie, będące autorami tych wersji danych [Nguyen 2002a].

W teorii consensusu można rozróżnić dwa podejścia do jego rozwiązania. Podejście aksjomatyczne, zasadzające się wg McMorrisa na wyznaczeniu własności, które powinny być spełnione przez metody consensusu, a następnie szukanie takich metod, które spełniają te własności. Drugie podejście, to podejście konstruktywne, gdzie problemy consensusu rozwiązuje się na dwóch poziomach: mikrostruktury oraz makrostruktury uniwersum obiektów  $U$ .

Przez mikrostrukturę zbioru  $U$  rozumie się strukturę jego elementów, np. porządek liniowy na pewnym zbiorze lub zbiór kandydatów. Należy przy tym dodać, że consensus nie musi mieć takiej samej mikrostruktury jak zbiór  $X$ . Przez makrostrukturę zbioru  $U$  rozumie się strukturę tego zbioru, np. relację preferencji lub funkcję odległości (lub podobieństwa) między obiektami ze zbioru  $U$ . Makrostrukturą zbioru  $U$  jest najczęściej pewna binarna relacja w zbiorze  $U$  lub funkcja odległości definiowana na podstawie jego mikrostruktury.

Teoria consensusu zajmuje się przede wszystkim dwoma typami problemów [Nguyen 2002a], pierwszy typ dotyczy poszukiwania ukrytej struktury obiektów, a drugi uzgadniania wiedzy dotyczącej tego samego obiektu. W pierwszym typie problemu wersje danych odzwierciedlają nieznaną strukturę, która powoduje, że występują trudności z pomiarem odległości między tymi strukturami, istnieje możliwość utraty informacji lub pojawienia się błędów w kryteriach ewaluacyjnych. W drugim typie problemu wersje danych różnią się między sobą, gdyż otrzymano je w różny sposób (na podstawie różnych obserwacji, eksperymentów itp.) oraz właściwa wersja nie jest znana, a problem consensusu może być rozpatrywany jako problem porządkowania (rangowania) alternatyw lub jako problem komitetu wyborczego [Nguyen 2002a].

W prezentowanym w niniejszej pracy modelu consensusu przyjęto jako obowiązujący paradygmat atrybutowy. W konsekwencji tego wyboru wersje danych zawartości consensusu są tworzone przez przestrzeń (uniwersum) krotek reprezentujących obiekty, zdarzenia, itp. Krotki reprezentowane są przez zbiór różnych atrybutów i ich wartości, które są zbiorem wartości elementarnych, innymi słowy w niniejszym modelu wykorzystywany jest tzw. wielowartościowy system informacyjny zdefiniowany w pracy [Pawlak 2001]. Paradygmat atrybutowy jest bardzo przydatny do opisywania świata rzeczywistego, a w szczególności obiektów tego świata. Wydaje się on również niezastąpiony w różnych modelach baz danych, od hierarchicznych po obiektowe, a także w metodach reprezentacji wiedzy.

### 5.2.1. Pojęcia podstawowe

W rozważaniach nad consensusem można założyć, że świat rzeczywisty może być opisany przez skończony zbiór atrybutów  $A$ . Niech  $V$  oznacza zbiór wartości elementarnych (dalej niepodzielnych) atrybutów  $V = \bigcup_{a \in A} V_a$ , gdzie  $V_a$  oznacza dziedzinę atrybutu  $a$ . Niech  $\Pi(V_a)$  oznacza zbiór podzbiorów zbioru  $V_a$ , natomiast  $\Pi(V_B) = \bigcup_{b \in B} (V_b)$  dla każdego  $B \subseteq A$ . W ramach modelu consensusu wykorzystywane będą następujące oznaczenia i–x [Nguyen 2002a] oraz xi–xii [Sobecki 2005]:

- i. Niech  $B \subseteq A$ , krotką typu  $B$  nazywamy funkcję  $r_B: B \rightarrow \Pi(V_B)$ , gdzie  $\forall_{b \in B}(r_b \subseteq V_b)$ .
- ii. Zbiór wszystkich krotek typu  $B$  oznacza się jako  $TYPE(B)$ .
- iii. Krotkę nazywamy elementarną, jeżeli wszystkie wartości atrybutów ze zbioru  $A$  są zbiorami pustymi bądź jednoelementowymi.
- iv. Zbiór krotek elementarnych typu  $B$  oznaczany jest jako  $E-TYPE(B)$ .
- v. Pusta krotka, której wszystkie wartości są zbiorami pustymi oznacza się symbolem  $\phi$ .
- vi. Częściowo pusta krotka z przynajmniej jedną wartością pustą oznaczana jest symbolem  $\theta$ .
- vii. Niepusty zbiór  $R$  krotek typu  $B$  nazywany jest relacją typu  $B$ , stąd  $R \subseteq TYPE(B)$ .
- viii. Suma dwóch krotek  $r$  i  $r'$  typu  $B$  jest krotką  $r''$  typu  $B$  ( $r'' = r \cup r'$ ), taką że  $\forall_{b \in B}(r''_b = r_b \cup r'_b)$ .
- ix. Iloczyn dwóch krotek  $r$  i  $r'$  typu  $B$  jest również krotką typu  $B$  ( $r'' = r \cap r'$ ) taką, że  $\forall_{b \in B}(r''_b = r_b \cap r'_b)$ .
- x. Mówimy, że krotka  $r$  zawarta jest w krotce  $r'$  ( $r \prec r'$ ), gdzie  $r, r' \in TYPE(B)$ , wtedy i tylko wtedy, gdy  $\forall_{b \in B}(r_b \subseteq r'_b)$ .
- xi. Mówimy, że relacja  $R$  typu  $B$  jest projekcją relacji  $R'$  typu  $B'$  ( $\Pi^B(R')$ ), wtedy i tylko wtedy, gdy  $B \subseteq B'$  i  $\forall_{b \in B}(r_b = r'_b)$ .
- xii. Mówimy, że zbiór atrybutów  $B$  jest kluczem relacji  $R$  ( $key(R)$ ) wtedy i tylko wtedy, gdy wartości tych atrybutów identyfikują w sposób unikatowy każdą krotkę relacji  $R$ .

### 5.3. Definicja systemu konfliktu

W ramach systemu konfliktu zakłada się, [Nguyen 2002a] i [Nguyen 2003a], że w świecie rzeczywistym agenty rozlokowane są w różnych lokacjach rozproszonego systemu, a przedmiotem ich „zainteresowania” są zdarzenia zachodzące w świecie rzeczywistym. Zadaniem agenta jest określenie wartości atrybutów opisujących te zdarzenia, przy czym samo zdarzenie określone jest przez pewną krotkę. Definicja ta jest pewną modyfikacją definicji przedstawionej w pracy [Nguyen 2003] polegającą na wprowadzeniu pojęcia kategorii  $c$  rekomendacji: demograficznej, kolaboratywnej i bazującej na zawartości [Sobecki 2005]. Drugą różnicą jest identyfikacja źródeł informacji konfliktowej, gdyż system konfliktu definiowany tutaj identyfikuje nie tylko różne agenty, ale w zależności od kategorii również różne zdarzenia zaobserwowane przez pojedynczego agenta. Wszystkie elementy systemu konfliktu zostały dalej opisane.

**Definicja 9**

System konfliktu pewnej kategorii  $c$  zdefiniowany jest jako czwórka:

$S^c = (A, X, P, Z)$ , gdzie:

- $A$  jest skończonym zbiorem atrybutów zawierającym atrybuty ze zbioru atrybutów  $T \subseteq A$  pozwalające zidentyfikować każde zdarzenie,
- $X$  jest skończonym zbiorem nośników konfliktów,  $X = \{\Pi(V_a): a \in A\}$ ,
- $P$  jest skończonym zbiorem relacji na nośnikach konfliktu ze zbioru  $X$ , każda relacja jest pewnego typu  $L$ , gdzie  $L \subseteq A$  i  $L$  zawiera atrybut lub atrybuty ze zbioru  $T$  pozwalające zidentyfikować każde zdarzenie,
- $Z$  jest skończonym zbiorem formuł logicznych, dla których system  $(X, P)$  jest modelem.

Zbiór atrybutów  $A$  należy traktować jako zestaw cech służących do opisywania określonej rzeczywistości, tak jak w przypadku reprezentacji profilu użytkownika w postaci krotki przedstawionej w punkcie 3.3.2.3. Zbiór nośników consensusu  $X$ , zawiera wszystkie wartości cech opisywanego świata, np. w przypadku systemu interakcyjnego są to z pewnością poszczególne wartości cech użytkowników, instancji interfejsu użytkownika lub konkretnych środowisk systemowo-sprzętowych, na którym implementowany jest interfejs użytkownika.

Relacje należące do zbioru  $P$  są sklasyfikowane w taki sposób, aby każda z nich reprezentowała podobne zdarzenia. W naszym przypadku obserwacje i zdarzenia dotyczą m.in. ustawień interfejsu użytkownika, stąd zbiór atrybutów  $A$  zawiera atrybuty opisujące interfejs użytkownika, atrybuty opisujące użytkownika (które można zaliczyć zarówno do danych użytkownika jak i użytkownika) oraz atrybuty pozwalające zidentyfikować odpowiednie zdarzenia.

**5.3.1. Profile konfliktu**

Sytuacja konfliktu dla danego systemu konfliktu  $S^c$  o kategorii  $c$  zawiera informacje o konkretnym konflikcie i zdefiniowana jest następująco:

**Definicja 10**

Sytuacja konfliktu  $cs^c$  dla danego systemu konfliktu  $S^c$  o kategorii  $c$  jest parą  $\langle P, Y \rightarrow B \rangle$ , gdzie  $P \in P$ ,  $Y \subseteq L \setminus T$ ,  $B \subseteq A \setminus T$ ,  $Y \cap B = \emptyset$  i  $r_Y \neq \theta$  dla każdej krotki  $r \in P$ .

Sytuacja konfliktu składa się z dwóch elementów: relacji  $P$ , zwanej podstawą konfliktu oraz przedmiotu consensusu  $Y \rightarrow B$ . Wyrażenie  $Y \rightarrow B$  oznacza, że w zaobserwowanych zdarzeniach występują różnice odnośnie do przyporządkowań wartościom atrybutów ze zbioru  $Y$  wartości atrybutów ze zbioru  $B$ . Celem wyboru consensusu jest znalezienie przyporządkowania każdej krotce typu  $Y$  dokładnie jednej krotki typu  $B$ .

Dla danej sytuacji  $cs^c$  określa się zbiór zdarzeń, które biorą udział w sytuacji konfliktu jako projekcję relacji  $P$  na zbiór atrybutów  $K$ ,  $\text{Event}(cs^c) = \Pi^K(P)$ , gdzie  $K \subseteq T$ ,

i  $K$  jest kluczem relacji  $P$ ,  $key(P)$ . Wówczas zbiór elementów podmiotowych (lub w skrócie podmiotów) jest zdefiniowany jako projekcja relacji  $P$  na zbiór atrybutów  $Y$ ,  $Subject(cs^c) = \Pi^Y(P)$ , gdzie  $Y \subseteq L \setminus T$ .

Należy zauważyć, że  $Event(cs^c)$  zawiera informacje pozwalające zidentyfikować sytuację konfliktu, natomiast  $Subject(cs^c)$  zawiera informacje, które są podmiotem sytuacji konfliktowej i determinują wartości atrybutów ze zbioru  $B$ . Mając zdefiniowane relacje: zdarzeń i podmiotów, można zdefiniować relację profili  $Profile(e)$ , jako relację zawierającą wiedzę pochodzącą ze zdarzeń na podmiotach  $e \in Subject(cs^c)$ , która zawiera wartości atrybutów ze zbioru  $K$  oraz zbioru  $B$ , czyli  $Profile(e) = \{r_{B \cup K} : (r \in P) \wedge (e = r_Y)\}$ . Konkretnie przykłady relacji  $Event(cs^c)$ ,  $Subject(cs^c)$  i  $Profile(e)$ , dla poszczególnych kategorii  $c$  przedstawiono w punktach 5.4 i 5.5.

### 5.3.2. Wyznaczenie consensusu dla profili użytkowników

Przedstawiona definicję consensusu i metody jego wyznaczenia opierająca się na ideach przedstawionych w pracy [Nguyen 2003a].

#### Definicja 11

Consensusem dla podmiotu  $e \in Subject(cs^c)$  sytuacji  $cs^c = \langle P, Y \rightarrow B \rangle$  jest krotka  $C(cs^c, e)$ , gdzie  $C(cs^c, e) \in TYPE(Y \cup B)$ , która spełnia formuły logiczne ze zbioru  $Z$  oraz przynajmniej jeden z poniższych postulatów:

P1. Zamkniętości wiedzy:  $C(cs^c, e)_B \prec \bigcup_{r \in Profile(e)} r_B$ .

P2. Zgodności wiedzy:  $\bigcap_{r \in Profile(e)} r_B \prec C(cs^c, e)_B$ .

P3. Warunek zgodności Condorceta dla wyboru:

Niech  $cs^{c'} = \langle P', Y \rightarrow B \rangle$ ,  $cs^{c''} = \langle P'', Y \rightarrow B \rangle$

jeżeli  $C(cs^{c'}, e)_B \cap C(cs^{c''}, e)_B$

to  $C(cs^c, e)_B = C(cs^{c'}, e)_B \cap C(cs^{c''}, e)_B$

gdzie  $cs^c = \langle P' \cup P'', Y \rightarrow B \rangle$ .

P4. Quasi-jednomyślności:

Jeżeli  $x \in Profile(e)$  i  $x_B \neq C(cs^c, e)_B$

to istnieje liczba naturalna  $n$  taka, że  $x = C(cs^{c'}, e)_B$ ,

gdzie  $cs^{c'}$  jest sytuacją konfliktową z tym samym podmiotem  $Y \rightarrow B$ , w którym  $Profile'(e) = Profile(e) \cup \{n * x\}$ .

Wymienione w definicji 11 postulaty mają następującą interpretację:

P1. Zamkniętość wiedzy, który mówi, że consensus powinien zawierać się w sumie elementów profilu.

P2. Zgodność wiedzy, który mówi, że część wspólna elementów profilu powinna zawierać się w consensusie.

P3. Warunek zgodności Condorceta dla wyboru [Taylor 1995], który mówi, że jeżeli dwa różne profile prowadzą do wspólnego consensusu, to consensus ten jest również zachowany dla sumy tych profili.

P4. Quasi-jednomyślności, który mówi, że jeżeli  $x$  nie jest consensusem pewnego profilu, to powinien on zostać consensusem tego profilu rozszerzonego  $n$ -krotnie o element  $x$ .

Twierdzenie 7 pozwala na wyznaczenie consensusu, który spełnia wszystkie postulaty P1–P4. Dowód tego twierdzenia jest modyfikacją twierdzenia zaprezentowanego w pracy [Nguyen 2000].

### Twierdzenie 7

Jeżeli mamy zdefiniowaną proporcjonalną funkcję odległości  $\delta$  między krotkami  $TYPE(B)$ , to dla danego podmiotu  $e$  i sytuacji konfliktu  $cs^c = \langle P, Y \rightarrow B \rangle$ , consensus  $C(cs^c, e)$ , który minimalizuje wyrażenie  $\sum_{r \in \text{Profile}(e)} \delta(r_B, C(cs^c, e)_B)$  spełnia wszystkie postulaty P1–P4 z definicji 11.

Podana własność consensusu zakłada, że wiarygodność każdego profilu jest jednakowa. Jednak w praktycznych sytuacjach można dopuścić różne wartości wiarygodności lub oceny przydatności określonego profilu. W takiej sytuacji należy zastosować pewne wagi przyjmujące wartości z przedziału  $[0, 1]$  w sumie odległości [Nguyen 2002a]:

$$\sum_{r \in \text{Profile}(e)} w(r_K) \delta(r_B, C(cs^c, e)_B), \quad (51)$$

gdzie:  $K$  jest kluczem relacji  $P$ , stąd  $r_K$  w sposób jednoznaczny identyfikuje każde zdanie, któremu można przypisać określoną wagę,  $w: V_K \rightarrow [0, 1]$ .

W pracy [Nguyen 2002a], wskazuje się, że dzięki odpowiedniemu zastosowaniu wag można uczynić profile podatnymi na consensus, nawet w sytuacji, gdy bez ich zastosowania nimi nie są.

W przypadku gdy wszystkie atrybuty w profilu są niezależne, wtedy wyznaczanie consensusu w profilu  $\text{Profile}(e)$  redukuje się do wyznaczenia consensusu dla każdego atrybutu w krotce  $TYPE(B)$ .

W zależności od mikrostruktury atrybutu, np. zbiorów jednoelementowych, zbiorów, przedziałów wartości i uporządkowań zbiorów oraz makrostruktury ich przestrzeni (definicji funkcji odległości) można wyróżnić różne algorytmy określenia consensusu. Najprostszą mikrostrukturą jest zbiór jednoelementowy o wartościach ze zbioru  $V_a$ , dla której można określić pewną funkcję odległości. Wtedy określenie consensusu profilu na podstawie twierdzenia 7 polega na wyborze takiego elementu, którego suma odległości do wszystkich elementów profilu jest minimalna. Powszechnie znaną metodą wyznaczania consensusu, która minimalizuje wyrażenie z twierdzenia 7 jest mediana [Nguyen 2002a].

Dla pozostałych wymienionych mikrostruktur algorytmy wyznaczenia consensusu są trochę bardziej skomplikowane. Zostały one przytoczone wraz z przykładami zastosowań w pracy [Nguyen 2003a].

## 5.4. Profil użytkownika

Profil użytkownika jest implementacją jego modelu w każdym systemie rekomendacyjnym. Tworząc profil użytkownika, należy określić kilka elementów przedstawionych w punkcie 3.3. Należą do nich: zawartość profilu, reprezentacja profilu, sposób inicjowania profilu, wykorzystane techniki uczenia profilu, sprzężenie zwrotne oraz techniki adaptacji profilu.

Ponieważ w prezentowanej metodzie wykorzystuje się filtrowanie demograficzne DF, podstawą wnioskowania są odpowiednie dane demograficzne, będące bardzo popularnym elementem danych użytkownika w wielu systemach rekomendacyjnych [Kobsa 2001]. W systemie *Megane* [Sobecki 2004b] wykorzystano dane demograficzne, które zawierają dane zapisowe (imię i nazwisko), dane geograficzne (liczba mieszkańców w miejscu zamieszkania), charakterystykę użytkownika (wiek, płeć i wykształcenie) oraz dane rejestracyjne (login i hasło). Oprócz danych demograficznych, dane użytkownika zawierają również preferencje użytkownika (dotyczące szczegółowości wyświetlanych informacji) i identyfikator klasy, do której został zaklasyfikowany dany użytkownik.

Profil użytkownika, oprócz danych użytkownika, zawiera również dane użytkownika [Kobsa 2001]. Głównym elementem danych użytkownika w systemie *Megane* są dane dotyczące konfiguracji interfejsu użytkownika oraz oceny satysfakcji z jego użytkowania, które są pewnym rodzajem danych obserwowalnych. Konfiguracja interfejsu użytkownika obejmuje wybór określonych wartości dla wybranych jego elementów: pasek opcji, tło, podkład muzyczny, dźwięki, menu główne, oraz wygląd poszczególnych sekcji systemu *Megane*: informacje, kolory, galeria oraz pliki. Ostatnim elementem danych użytkownika jest ocena satysfakcji z użytkowania tego interfejsu.

Profil użytkownika w systemie *Megane* reprezentowany jest za pomocą krotki opisanej w punkcie 3.3.2.3. Metoda reprezentacji nazywana jest w ramach teorii consensusu określeniem mikrostruktury zbioru elementów [Nguyen 2003a]. Aby zdefiniować krotkę należy określić zbiór skończony atrybutów  $A$  oraz zbiór wartości tych atrybutów  $V$ . Zbiór atrybutów  $A$  dla systemu *Megane* można podzielić na dwie części, atrybuty danych użytkownika  $A^u = \{\text{login}, \text{hasło}, \text{imię}, \text{nazwisko}, \text{wiek}, \text{płeć}, \text{wykształcenie}, \text{liczba\_mieszkańców\_w\_miejscu\_zamieszkania}, \text{preferowany\_typ\_informacji}, \text{identyfikator\_klasy}\}$  oraz atrybuty danych użytkownika  $A^i = \{\text{pasek\_opcji}, \text{tło}, \text{podkład\_muzyczny}, \text{dźwięki}, \text{głośność\_podkładu}, \text{głośność\_dźwięków}, \text{menu\_główne}, \text{sekcja\_informacje}, \text{sekcja\_kolory}, \text{sekcja\_galeria}, \text{sekcja\_pliki}, \text{satysfakcja}\}$ , wówczas  $A = A^u \cup A^i$ . Zbiory wartości atrybutów określone są w następujący sposób przedstawiono w tabeli 5.

Mając określony zbiór atrybutów  $A$  i zbiór wartości tych atrybutów  $V$ , można zdefiniować krotkę  $r$ , jako funkcję  $r: A \rightarrow V$ , gdzie  $\forall_{a \in A} (r(a) \in V_a)$ . W systemie *Megane* profil użytkownika reprezentowany jest jako krotka  $r$ .



Tabela 5. Określenie zbiorów wartości atrybutów

| Lp. | Nazwa zbioru                                                                                                                                                   | Zbiór wartości                                                                                                                  | Opis                                                                                                                                                                                    |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1   | $V_{\text{login}}$                                                                                                                                             | $\{v: v=a_1a_2\dots a_n, \text{ gdzie } a_1, a_2, \dots, a_n \in \{a, \dots, z; A, \dots, Z\} \text{ oraz } 1 \leq n \leq 10\}$ | Ciąg znaków o maksymalnej długości 10 elementów                                                                                                                                         |
| 2   | $V_{\text{hasło}}$                                                                                                                                             | $\{v: v=a_1a_2\dots a_n, \text{ gdzie } a_1, a_2, \dots, a_n \in \{a, \dots, z; A, \dots, Z\} \text{ oraz } 1 \leq n \leq 10\}$ | Ciąg znaków o maksymalnej długości 10 elementów                                                                                                                                         |
| 3   | $V_{\text{imię}}$                                                                                                                                              | $\{v: v=a_1a_2\dots a_n, \text{ gdzie } a_1, a_2, \dots, a_n \in \{a, \dots, z; A, \dots, Z\} \text{ oraz } 1 \leq n \leq 10\}$ | Ciąg znaków o maksymalnej długości 15 elementów                                                                                                                                         |
| 4   | $V_{\text{nazwisko}}$                                                                                                                                          | $\{v: v=a_1a_2\dots a_n, \text{ gdzie } a_1, a_2, \dots, a_n \in \{a, \dots, z; A, \dots, Z\} \text{ oraz } 1 \leq n \leq 15\}$ | Ciąg znaków o maksymalnej długości 15 elementów                                                                                                                                         |
| 5   | $V_{\text{wiek}}$                                                                                                                                              | $\{mniej\_niż\_18, od\_18\_do\_25, od\_25\_do\_50, ponad\_50\};$                                                                | Wiek osoby może należeć do jednego z czterech przedziałów: poniżej 18 roku życia, między 18 a 25 rokiem życia, między 25 a 50, oraz ostatnia powyżej 50 lat                             |
| 6   | $V_{\text{płeć}}$                                                                                                                                              | $\{kobieta, mężczyzna\}$                                                                                                        | Płeć osoby jest określona przez zbiór dwuelementowy: kobieta lub mężczyzna                                                                                                              |
| 7   | $V_{\text{wykształcenie}}$                                                                                                                                     | $\{\text{podstawowe, średnie, wyższe}\}$                                                                                        | Wykształcenie osoby określone jest za pomocą zbioru zawierającego trzy elementy: podstawowe, średnie lub wyższe                                                                         |
| 8   | $V_{\text{liczba\_mieszkańców\_w\_miejscu\_zamieszkania}}$                                                                                                     | $\{mniej\_niż\_100.000, od\_100.000\_do\_500.000, od\_500.000\_do\_1.000.000, powyżej\_1.000.000\}$                             | Liczba mieszkańców w miejscowości jest podzielona na cztery przedziały: mniejszej niż 100 000, od 100 000 do 500 000, od 500 000 do 1 000 000, powyżej 1 000 000.                       |
| 9   | $V_{\text{preferowany\_typ\_informacji}}$                                                                                                                      | $\{\text{ogólne, szczegóły\_techniczne}\}$                                                                                      | Jest to zbiór dwuelementowy, który określa jaki typ informacji preferuje użytkownik. Może to być dwojaki rodzaj informacji: ogólne lub zawierający szczegóły techniczne                 |
| 10  | $V_{\text{identyfikator\_klasy}}$                                                                                                                              | $\{1, 2, 3, 4\}$                                                                                                                | Poszczególne numery identyfikują odpowiednią klasę                                                                                                                                      |
| 11  | $V_{\text{pasek\_opcji}}$                                                                                                                                      | $\{1, 2, 3, 4\}$                                                                                                                | Poszczególne numery reprezentuje typ paska posortowany od najbardziej nowoczesnego do najbardziej tradycyjnego                                                                          |
| 12  | $V_{\text{tło}} = V_{\text{menu\_główna}} = V_{\text{sekcja\_informacje}} = V_{\text{sekcja\_kolory}} = V_{\text{sekcja\_galeria}} = V_{\text{sekcja\_pliki}}$ | $\{1, 2, 3, 4, 5\}$                                                                                                             | Poszczególne numery reprezentują typ wyglądu posortowany od najbardziej nowoczesnego do najbardziej tradycyjnego                                                                        |
| 13  | $V_{\text{podkład\_muzyczny}} = V_{\text{dźwięki}}$                                                                                                            | $\{1, 2, 3, 4, 5\}$                                                                                                             | Poszczególne numery reprezentują typ podkładu lub dźwięków posortowany od najbardziej nowoczesnego do najbardziej tradycyjnego                                                          |
| 14  | $V_{\text{głośność\_podkład\_muzyczny}} = V_{\text{głośność\_dźwięki}}$                                                                                        | $\{0, 1, \dots, 100\}$                                                                                                          | Poszczególne numery reprezentują głośność podkładu muzycznego i dźwięków, od wyłączonego dźwięku przez najcichszą do najgłośniejszej                                                    |
| 15  | $V_{\text{satisfakcja}}$                                                                                                                                       | $\{0, 1, 2, 3, 4\}$                                                                                                             | Poszczególne numery reprezentują satysfakcję z korzystania z interfejsu nadaną przez użytkownika posortowaną od poziomu najmniej satysfakcjonującego do najbardziej satysfakcjonującego |

Inicjowanie profilu użytkownika odbywa się w ręcznie, przy czym w części profilu zawierającego dane użytkownika, oprócz identyfikatora klasy, oraz oceny satysfakcji wyłącznie w ten sposób. Natomiast atrybuty konfiguracji interfejsu mogą być inicjowane zarówno ręcznie jak i na drodze wnioskowania stereotypowego, na podstawie wyznaczonego identyfikatora klasy. Inicjowanie profilu realizowane jest na drodze rejestracji, patrz rys. 17, gdzie zaprezentowano przykładowe okno rejestracji w systemie *Megane* w wersji w języku angielskim.

Użytkownik może zmodyfikować postać interfejsu przez opcję personalizacji, przykładowe okno z działania aplikacji w wersji angielskiej przedstawiono na rysunku 18.

W systemie *Megane* wykorzystuje się techniki uczenia profilu polegające zarówno na klasyfikacji, jak i na grupowaniu. W celu przeprowadzenia klasyfikacji i grupowania należy określić odpowiednio funkcje odległości i podobieństwa.

Definicja funkcji odległości nazywana jest w ramach teorii consensusu określeniem makrostruktury zbioru elementów [Nguyen 2003a]. Ze względu na unikatowe cechy profilu dla każdej implementacji, definicja odległości powinna uwzględniać tę specyfikę.

Funkcja odległości między wartościami pewnych atrybutów  $a \in D \subset A$  jest zdefiniowana jako funkcja  $\delta^a: V_a \times V_a \rightarrow [0,1]$ . Z powodu braku istotności dla rekomendacji interfejsów użytkownika nie jest określona odległość między następującymi atrybutami:  $\{\text{login}, \text{hasło}, \text{imie}, \text{nazwisko}\}$ . Funkcje odległości dla każdego atrybutu powinny zostać określone indywidualnie przez projektanta systemu, jednak nie muszą one spełniać warunku metryki. Wartości funkcji odległości mogą zostać podane w sposób proceduralny lub też, dla niektórych atrybutów, poszczególne jej wartości mogą zostać podane dla wszystkich par ich wartości.

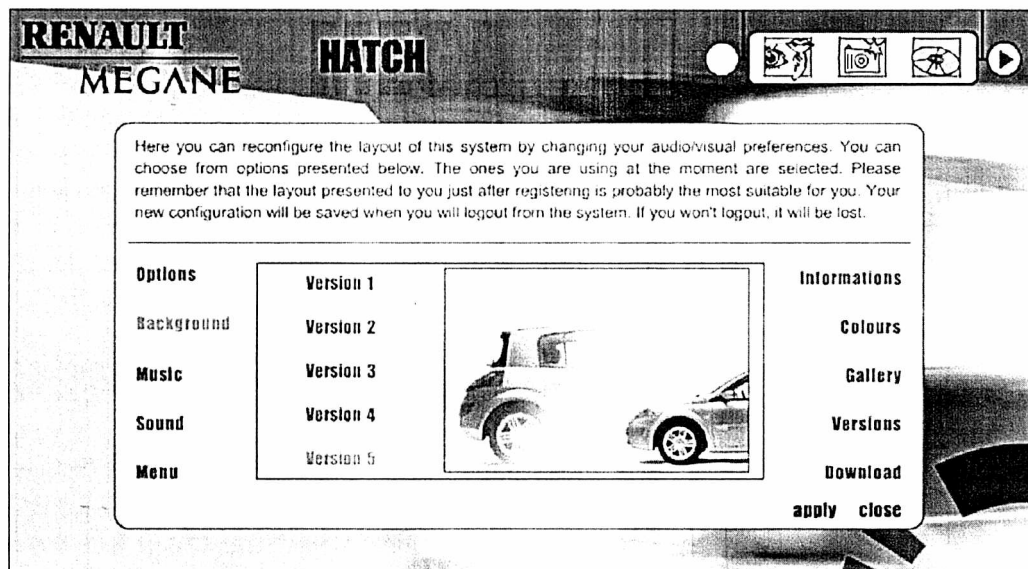
RENAULT MEGANE HATCH

Here you have the possibility to register. In order to register, you must fill in all the required fields (except for the name and surname - they are optional) and answer all the questions. Registering is necessary to fully use this system. After registering, the system will adjust to your needs and preferences.

|           |                                   |                                             |                                               |
|-----------|-----------------------------------|---------------------------------------------|-----------------------------------------------|
| Login:    | <input type="text"/>              | Education:                                  | <input type="text" value="Primary"/>          |
| Password: | <input type="text"/>              | How many city dwellers does your city have? | <input type="text" value="&lt;100.000"/>      |
| Name:     | <input type="text"/>              | The information you are interested in is:   | <input type="text" value="rather technical"/> |
| Surname:  | <input type="text"/>              |                                             |                                               |
| Age:      | <input type="text"/>              |                                             |                                               |
| Sex:      | <input type="text" value="Male"/> |                                             |                                               |

close

Rys. 17. Okno rejestracji użytkownika w systemie Megane [Sobecki 2004b]



Rys. 18. Okno personalizacji interfejsu systemu Megane [Sobecki 2004b]

Wartości dla poszczególnych atrybutów są posortowane, więc można w prosty sposób wyznaczyć odległości między nimi [Sobecki 2004b]. Wśród zbiorów wartości atrybutów, które traktujemy jako zbiory uporządkowane, można wyróżnić zbiory dwu-, cztero- i pięcioelementowe. Odległości między różnymi wartościami tych zbiorów określone są na podstawie wzoru:

$$\delta^a(v_a^i, v_a^j) = \frac{|i-j|}{\text{card}(V_a)-1}, \quad (52)$$

gdzie:  $v_a^i, v_a^j$  są odpowiednio  $i$ -tą i  $j$ -tą wartością z uporządkowanego zbioru  $V_a$  dla  $i \neq j$ .

Po określeniu odległości dla poszczególnych wartości atrybutów można określić sposób wyznaczenia odległości między krotkami  $TYPE(D)$ :  $r_i, r_j$ . Odległość między krotkami  $r_i, r_j$  można zdefiniować jako sumę odległości:

$$\delta(r_i, r_j) = \frac{1}{\text{card}(D)} \sum_{a \in D} \delta^a(r_i(a), r_j(a)). \quad (53)$$

Drugą z możliwości jest wykorzystanie pierwiastka z sumy kwadratów:

$$\delta(r_i, r_j) = \frac{1}{\text{card}(D)} \sqrt{\sum_{a \in D} [\delta^a(r_i(a), r_j(a))]^2}. \quad (54)$$

Trzecia możliwość wyznaczenia odległości między krotkami  $r_i, r_j$  polega na przemnożeniu odległości dla poszczególnego atrybutu przez odpowiedni współczynnik  $w$ :  $A \rightarrow [0,1]$ :

$$\delta(r_i, r_j) = \frac{1}{\text{card}(D)} \sum_{a \in D} w(a) \delta^a(r_i(a), r_j(a)). \quad (55)$$

W systemie *Megane* zastosowano najprostszą wersję wyrażoną wzorem (53) wyznaczania odległości między krotkami reprezentującymi dane użytkownika.

Do wyznaczenia odległości między profilem użytkownika a centroidami reprezentującymi poszczególne klasy wykorzystuje się następujący zbiór atrybutów:

$D = \{\text{wiek}, \text{płeć}, \text{wysztalcenie}, \text{liczba\_mieszkańców\_w\_miejscu\_zamieszkania}, \text{preferowany\_typ\_informacji}\}.$

Po wyznaczeniu odległości między krotką reprezentującą bieżącego użytkownika a wszystkimi centroidami, identyfikuje się centroid, do którego odległość jest najmniejsza i nadaje się wartość identyfikatora klasy  $k$ , którą reprezentuje ten centroid:  $r_{\text{identyfikator\_klasy}} = k$ .

W systemie *Megane* jest również wykorzystywana druga obok klasyfikacji technika uczenia profilu, a mianowicie – grupowanie. Grupowanie realizowane jest na podstawie algorytmu Dattoli, który został opisany w punkcie 3.3.4.2. Algorytm ten wyznacza również nowe centroidy utworzonych grup. Istotnym parametrem tego algorytmu jest liczba grup, które należy utworzyć. W przypadku systemu *Megane* została ona ustalona na 4, czyli mniej więcej tyle, ile różnych wartości mają atrybuty określające wygląd interfejsu dla różnych opcji systemu *Megane*. Grupowanie realizowane jest na podstawie funkcji podobieństwa, która jak to wynika z twierdzenia 1 może zostać wyznaczona z wykorzystaniem funkcji odległości:  $\sigma(r_i, r_j) = 1 - \delta(r_i, r_j)$ , która bazuje na funkcji odległości wykorzystywanej w klasyfikacji. Efektem grupowania jest wyznaczenie nowych centroidów oraz modyfikacja wartości  $r_{\text{identyfikator\_klasy}}$  dla wszystkich zarejestrowanych użytkowników. Algorytm, oprócz czterech klas, identyfikuje obiekty izolowane, którym nie zostaje na tym etapie przydzielona żadna klasa i wartość atrybutu  $\text{identyfikator\_klasy}$  pozostaje zbiorem pustym.

W systemie wykorzystywane jest sprzężenie zwrotne, w postaci pozyskania od użytkownika oceny satysfakcji z użytkowania interfejsu użytkownika, którą nadaje sam użytkownik manualnie. Zastosowana ocena ma pięć wartości całkowitoliczbowych od 0 do 4, gdzie 0 reprezentuje zupełnie niesatysfakcjonujący, a 4 najbardziej satysfakcjonujący interfejs użytkownika.

## 5.5. Wykorzystanie profilu użytkownika

Wykorzystanie profilu użytkownika jest określone zastosowaną metodą filtrowania informacji. W systemie *Megane* stosuje się hybrydową metodą filtrowania, która jest złożeniem metody demograficznej i kolaboratywnej.

Filtrowanie demograficzne stosowane jest jedynie na początkowym etapie działania systemu w celu przewyciężenia problemu filtrowania kolaboratywnego polegającego na ograniczonej liczbie użytkowników, którzy dostarczyli oceny (ang. *sparsity problem*).

Natomiast, gdy w systemie zostanie zarejestrowana odpowiednia liczba użytkowników, inicjowane są procedury filtrowania kolaboratywnego.

DF wykorzystuje wnioskowanie stereotypowe, na podstawie wartości identyfikatora klasy  $r_{\text{identyfikator\_klasy}}$ , która została nadana na etapie klasyfikacji profilu na podstawie danych demograficznych. Podstawą klasyfikacji jest funkcja odległości zdefiniowana w przestrzeni profili użytkownika. Rezultatem działania wnioskowania stereotypowego jest nadanie odpowiednich wartości atrybutom odpowiadającym za konfigurację interfejsu użytkownika  $B \subset A^i \subset A$ , gdzie  $B = A^i \setminus \{\text{satysfakcja}\} = \{\text{pasek\_opcji}, \text{tło}, \text{podkład\_muzyczny}, \text{dźwięki}, \text{głośność\_podkładu}, \text{głośność\_dźwięków}, \text{menu\_główne}, \text{sekcja\_informacje}, \text{sekcja\_kolory}, \text{sekcja\_galeria}, \text{sekcja\_pliki}\}$ . Co prawda w DF występującym w systemie *Megane* nie są wykorzystywane metody consensusu, jednak można byłoby je zastosować, gdyby stereotypy dla poszczególnych klas budował więcej niż jeden ekspert i zaistniałaby konieczność wyznaczenia consensusu między opiniami wielu ekspertów co do konfiguracji interfejsu.

Zaimplementowane w systemie *Megane* filtrowanie kolaboratywne (CF) wykorzystuje metody consensusu opisane w punkcie 4.1.3. Aby mechanizmy CF zostały zastosowane w systemie, powinna zostać zarejestrowana odpowiednio liczna grupa użytkowników, którzy oceniają wykorzystywane interfejsy użytkownika, które uprzednio mieli okazję spersonalizować.

Do obrazowania działania metod consensusu w systemie *Megane* zostaną opisane poszczególne elementy systemu konfliktowego i metody wyznaczania consensusu. Tak więc system konfliktowy określony w definicji 9 ma następującą postać  $S^{CF} = (A, X, P, Z)$ , gdzie:

- $A$  jest zbiorem atrybutów systemu *Megane*, natomiast zbiorem atrybutów, który pozwala zidentyfikować każde zdarzenie zbiorów atrybutów  $T = \{\text{login}\} \subseteq A$ ,
- $X$  jest skończonym zbiorem nośników konfliktów,  $X = \{\Pi(V_a) : a \in A\}$ , dla każdego  $V_a$  określonego dla systemu *Megane*,
- $P$  jest skończonym zbiorem relacji na nośnikach konfliktu ze zbioru  $X$ , zawierających dane o wszystkich zarejestrowanych użytkownikach, ich danych użytkownika i użytkowania, oprócz tych, dla których wartość atrybutu *identyfikator\_klasy* jest zbiorem pustym, identyfikatorem każdego zdarzenia jest  $T \subseteq L$ , gdzie  $L = \{\text{identyfikator\_klasy}, \text{login}\}$  i  $L \subseteq A$ ,
- $Z$  zawiera następującą formułę:  $\forall_{P \in P} \forall_{r \in P} (r \neq \emptyset)$ ,

Sytuacja konfliktowa  $cs^{CF}$ , określona w definicji 10 dla danego systemu konfliktowego  $S^{CF}$ , jest parą  $\langle P, Y \rightarrow B \rangle$ , gdzie  $P \in P$  jest relacją  $\text{TYPE}(\{\text{identyfikator\_klasy}, \text{login}\} \cup A^i)$ ,  $Y$  jest zbiorem atrybutów, które mają wpływ na zbiór atrybutów dotyczący ustawienia interfejsu, tzn. identyfikatorem klasy,  $Y = \{\text{identyfikator\_klasy}\}$ ,  $B = A^i \setminus \{\text{satysfakcja}\}$ ,  $Y \cap B = \emptyset$  i  $r_Y \neq \emptyset$  (co oznacza, że identyfikator klasy nie może być pusty) dla każdej krotki  $r \in P$ .

Mając zdefiniowany system konfliktowy  $S^{CF}$  i sytuację konfliktową  $cs^{CF}$ , można przystąpić do określenia zdarzenia,  $\text{Event}(cs^{CF}) = \Pi^K(P)$ , gdzie kluczem relacji  $P$  jest  $K = \{\text{login}\}$ , stąd zbiór zdarzeń jest zbiorem wartości atrybutu *login* z relacji  $P$ , czyli

takich użytkowników, dla których identyfikator klasy nie jest pusty. Natomiast podmiot relacji dany jest zbiorem:  $\text{Subject}(cs^{CF}) = \Pi^Y(P)$ , gdzie  $Y = \{\text{identyfikator\_klasy}\}$ . Mając zdefiniowane zbiory: zdarzeń i podmiotów, można zdefiniować zbiór profili  $\text{Profile}(e)$ , jako zbiór zawierający wiedzę pochodzącą ze zdarzeń na podmiotach  $e \in \text{Subject}(cs^c)$ , który zawiera wartości atrybutów ze zbioru  $K$  oraz zbioru  $B$ , który zawiera informacje dotyczące konfiguracji interfejsu użytkownika –  $(B \cup K)$ , czyli  $\text{Profile}(e) = \{r_{B \cup K} : (r \in P) (e \prec r_Y)\}$ .

Mając określone wszystkie komponenty, można przystąpić do wyznaczenia consensusu  $C(cs^{CF}, e)$  dla poszczególnych identyfikatorów klas, na podstawie zawartości odpowiednich profili. Z twierdzenia 7, z dalszymi modyfikacjami, wynika, że consensus  $C(cs^{CF}, e)$ , który minimalizuje wyrażenie (51)  $\sum_{r \in \text{Profile}(e)} w(r_{\{\text{login}\}}) \delta(r_B, C(cs^c, e)_B)$ ,

gdzie waga wyznaczona jest w następujący sposób:  $w(r_{\{\text{login}\}}) = \frac{r_{\{\text{satisfakcja}\}}}{4}$ , gdzie  $r \in P$  oraz 4 w mianowniku oznacza maksymalną wartość satysfakcji, spełnia postulaty P1–P4.

Z założenia atrybuty ze zbioru  $B$  są niezależne, stąd można wyznaczyć consensus osobno dla każdego z tych atrybutów, a następnie consensusem jest krotka  $\text{TYPE}(B)$ , która zawiera poszczególne wartości. Jedną z najprostszych metod wyznaczenia consensusu, który minimalizuje ważoną sumę odległości między consensusem a wszystkimi elementami profilu, jest ważona mediana (patrz punkt 5.3.2), którą z powodzeniem można zastosować dla wszystkich wartości atrybutów ze zbioru  $B$ , gdyż wartości wszystkich atrybutów ze zbioru  $B$  można uporządkować.

Po wyznaczeniu consensusu dla każdego identyfikatora klasy dysponujemy nowymi stereotypami konfiguracji interfejsu, które przyporządkowywane są wszystkim nowym użytkownikom, którzy zostali zaklasyfikowani do określonej klasy, realizując w ten sposób drugą metodę filtrowania – CF.

W ramach opisu taksonomii systemów rekomendacyjnych pozostałymi elementami metod wykorzystania profilu, oprócz metody filtrowania, są techniki dopasowania rekomendowanego elementu do profilu użytkownika oraz dopasowanie między profilami. W przypadku pierwszego elementu w systemie *Megane* stosowana jest klasyfikacja, gdyż rekomendacja interfejsu dokonywana jest na podstawie przydzielenia do klasy użytkowników, której przypisane są odpowiednie rekomendacje. Natomiast w przypadku dopasowania między profilami, wykorzystuje się metodę grupowania Dattoli, która wyznacza grupę użytkowników, na podstawie których, z wykorzystaniem metod consensusu, wyznaczona jest odpowiednia predykcja.

## 5.6. Badania eksperymentalne systemu *Megane*

W celu weryfikacji zaproponowanej metody system *Megane* został zaimplementowany, a następnie przetestowany z udziałem użytkowników. System został zaimplementowany w architekturze trójwarstwowej, która pozwala na separację trzech głównych elementów dynamicznej aplikacji webowej: warstwę prezentacji, warstwę logiki bizne-

owej i warstwę danych. Wykorzystując taką architekturę systemu, uzyskuje się wysoką skalowalność systemu. Dodatkowo, każda warstwa zaimplementowana jest w innej technologii, co pozwala, w razie potrzeby, na jej wymianę bez zbędnych strat (np. warstwę danych zaimplementowaną w MySQL można zaimplementować w technologii MS SQL Server). Również cała logika biznesowa została skoncentrowana w jednej warstwie zaimplementowanej w PHP [Welling 2003], co pozwala na w miarę łatwą jej modyfikację i zmianę technologii, co byłoby konieczne w przypadku tysięcy użytkowników, gdyż technologia PHP nie pozwala, tak jak np. programy napisane w języku C, na szybkie obliczenia numeryczne. Skalowalność wspierana jest również dzięki zastosowaniu języka XML do wymiany danych między poszczególnymi warstwami.

Warstwa prezentacji, w celu uwydatnienia walorów multimedialnych, została zaimplementowana w Macromedia Flash MX (obecnie Adobe). Technologia ta od wielu lat doceniana jest przez projektantów i programistów realizujących zaawansowane webowe systemy multimedialne. Flash pozwala na wygodne integrowanie różnych mediów: tekstu, grafiki, dźwięku, animacji i wideo w jednej aplikacji.

### 5.6.1. Przebieg eksperymentu

Badanie efektywności systemu *Megane* zostało przeprowadzone z udziałem 75 użytkowników w warunkach kontrolowanych. Wszyscy uczestnicy eksperymentu byli studentami studiów magisterskich i podyplomowych na Wydziale Informatyki i Zarządzania Politechniki Wrocławskiej. Dzięki udziałowi studentów ze studiów podyplomowych udało się uzyskać większe zróżnicowanie użytkowników, zarówno pod względem wieku, jak i płci, niż w przypadku udziału wyłącznie studentów studiów dziennych. Eksperyment został przeprowadzony w trzech etapach.

W pierwszym etapie użytkownicy zarejestrowali się, a następnie zostali zaklasyfikowani do jednej z czterech klas, przy czym przyjęto, iż przy odległości od każdego centroidu większej niż 0,4 użytkownik nie został przyporządkowany do żadnej klasy. Następnie w zależności od klasy, do której użytkownik został przydzielony, przyporządkowano mu odpowiedni interfejs, przy czym w przypadku gdy użytkownik nie został przydzielony do żadnej klasy, przyporządkowano mu interfejs domyślny. Na tym etapie użytkownicy nie mogli personalizować interfejsu. Na zakończenie pracy z systemem, co następowało po zapoznaniu się z wszystkimi podstawowymi jego opcjami, użytkownicy byli proszeni o wypełnienie elektronicznej ankiety satysfakcji, oceniającej sześć aspektów użyteczności, tj. zawartość i strukturę informacyjną, zawartość multimedialną, interakcję mechanizmów rejestracji i personalizacji, w skali ocen od 1 do 10.

Każdą ocenę użytkownik mógł uzupełnić o pewien komentarz. Celem realizacji tego etapu było sprawdzenie efektywności filtrowania demograficznego (DF) oraz dostarczenie danych porównawczych z filtrowaniem kolaboratywnym (CF).

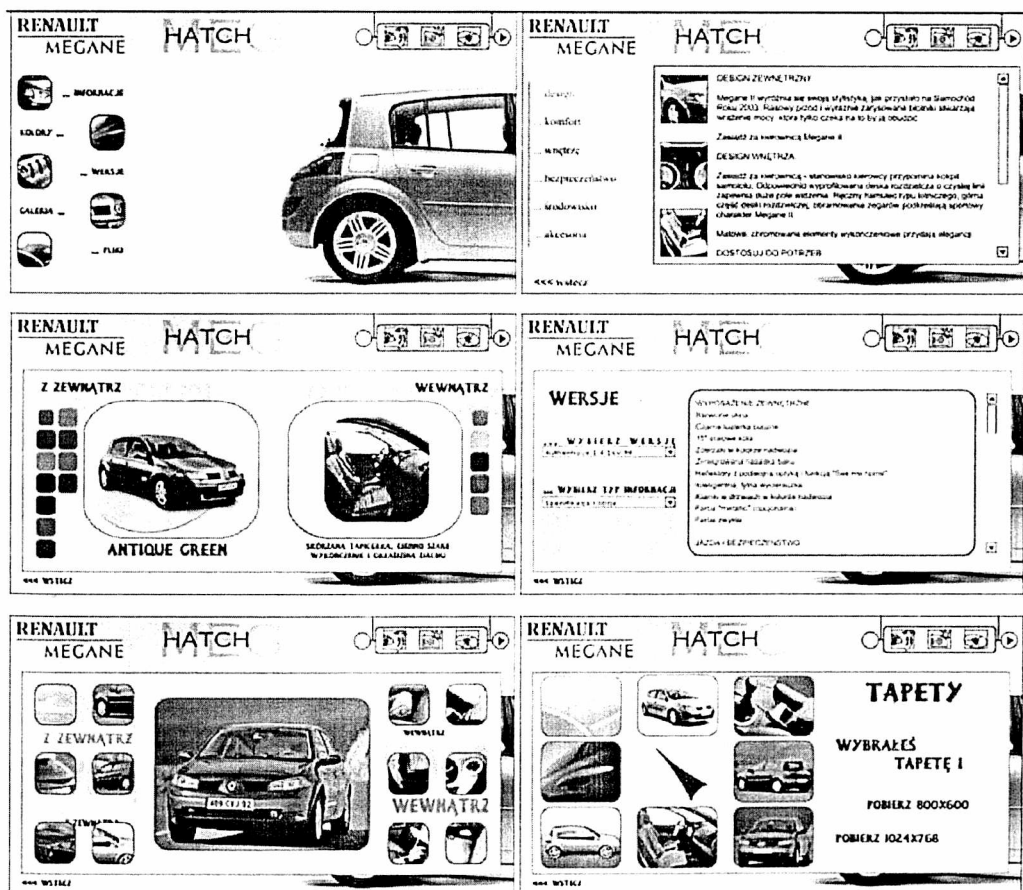
W tabeli 6 podano wartości poszczególnych atrybutów dla startowych centroidów wykorzystywanych w DF oraz procentowy udział użytkowników – uczestników pierwszego etapu eksperymentu. Wartości zostały dobrane, biorąc pod uwagę ich rozkład w całym społeczeństwie zamieszkującym nasz kraj w czasie realizacji eksperymentu (2003 r.).



Tabela 6. Wartości centroidów dla DF oraz procentowy udział użytkowników

| Klasa         | Procentowy udział | Płeć      | Wiek        | Wykształcenie | Liczba mieszkańców w miejscu zamieszkania | Preferowany typ informacji |
|---------------|-------------------|-----------|-------------|---------------|-------------------------------------------|----------------------------|
| 1             | 49                | mężczyzna | od_18_do_25 | podstawowe    | od_500.000_do_1.000.000                   | szczegóły techniczne       |
| 2             | 0                 | mężczyzna | ponad_50    | podstawowe    | od_500.000_do_1.000.000                   | szczegóły techniczne       |
| 3             | 30                | kobieta   | od_25_do_50 | średnie       | od_100.000_do_500.000                     | ogólne                     |
| 4             | 3                 | kobieta   | ponad_50    | wyższe        | powyżej_1.000.000                         | ogólne                     |
| Izolowana (0) | 18                | —         | —           | —             | —                                         | —                          |

Na rysunku 19 przedstawiono realizację graficzną interfejsu użytkownika rekomendowanego dla klasy 1.



Rys. 19. Zrzuty ekranowe wybranych opcji systemu Megane dla klasy 1

W drugim kroku nowa grupa użytkowników została poproszona o zarejestrowanie i spersonalizowanie swoich interfejsów zgodnie z własnymi preferencjami. Na koniec pracy użytkownicy zostali poproszeni o ogólną ocenę systemu w skali od 1 do 4.

W tabeli 7 przedstawiono zawartość profilu wybranych użytkowników, uczestników drugiego etapu eksperymentu; nie zostały umieszczone imiona i nazwiska, aby zachować anonimowość uczestników eksperymentu.

Po zakończeniu tego etapu profile wszystkich biorących w tym etapie użytkowników zostały pogrupowane, centroidy zostały wyznaczone metodą Dattoli, a następnie metodą consensusu zostały przygotowane dla poszczególnych klas rekomendacje konfiguracji interfejsu. W tabeli 8 podano wartości poszczególnych atrybutów dla centroidów po drugim etapie eksperymentu oraz procentowy udział użytkowników – uczestników pierwszego etapu eksperymentu. Następnie dla wszystkich użytkowników zaklasyfikowanych do poszczególnych klas metodą consensusu tworzone są nowe rekomendacje, przy czym użytkownicy, którzy nie zostali zaklasyfikowani do żadnej klasy, nie są brani pod uwagę. Rekomendacja domyślna dla obiektów izolowanych pozostaje bez zmian.

W trzecim etapie, podobnie jak w pierwszym, użytkownicy rejestrują się w systemie oraz na podstawie wprowadzonych danych użytkownika, obejmujących dane demograficzne, zostają przyporządkowani do klas o najmniejszej wartości funkcji odległości (jeżeli nie przekracza ona wartości 0,4), a następnie zostaje im przydzielona odpowiednia konfiguracja interfejsu. Tak jak w pierwszym etapie eksperymentu, użytkownicy nie mogli personalizować interfejsu. Na zakończenie pracy z systemem użytkownicy ponownie byli proszeni o wypełnienie elektronicznej ankiety takiej samej jak w etapie pierwszym.

### 5.6.2. Wyznaczenie efektywności rekomendacji interfejsu użytkownika

Badanie efektywności rekomendacji interfejsu użytkownika oparto na ankiecie, którą wypełniali użytkownicy po pierwszym i trzecim etapie eksperymentu. Ankieta ta obejmowała następujące aspekty:

1. Merytoryczna zawartość treści prezentowanych przez system obejmująca: teksty, grafiki, animacje, sekwencje dźwięku i podkładu muzycznego; oraz ocena samej tematyki systemu informacyjnego.

2. Jakość prezentowanych mediów, zwłaszcza: wielkość, kolor i krój czcionki, czytelność tekstu, jakość oraz rozmiar grafiki, i animacji, możliwość sterowaniem sekwencjami dźwięku i podkładem muzycznym.

3. Struktura informacji, zwłaszcza: spójność prezentowanych informacji oraz użycie odpowiedniej terminologii.

4. Ocena elementów interaktywnych, zwłaszcza: ich semantyka, reakcja elementów interaktywnych na działanie użytkownika, walory estetyczne interfejsu użytkownika oraz kompozycja graficzna całości.

Tabela 7. Zawartość profilu użytkownika dla wybranych użytkowników po drugim etapie

|                    |                          |                           |                           |                       |                          |                    |                           |                          |                       |
|--------------------|--------------------------|---------------------------|---------------------------|-----------------------|--------------------------|--------------------|---------------------------|--------------------------|-----------------------|
| Login              | Pawełg                   | Jazzman                   | Jakecz                    | Dyk                   | Allisza                  | Alusia             | Podymak                   | Student                  | Pk                    |
| Płeć               | M                        | M                         | K                         | M                     | K                        | K                  | K                         | M                        | M                     |
| Wiek               | od_18_do_25              | od_18_do_25               | od_18_do_25               | od_18_do_25           | od_18_do_25              | ponad_50           | od_25_do_50               | od_25_do_50              | od_25_do_50           |
| Wykształcenie      | Średnie                  | Średnie                   | Średnie                   | Średnie               | Średnie                  | Wyższe             | Wyższe                    | Wyższe                   | Wyższe                |
| L.mieszk. w msc.z. | mniej_niż<br>_100.000    | od_100.000<br>_do_500.000 | od_100.000<br>_do_500.000 | mniej_niż<br>_100.000 | mniej_niż<br>_100.000    | pnad<br>_1.000.000 | od_100.000<br>_do_500.000 | mniej_niż<br>_100.000    | mniej_niż<br>_100.000 |
| Pref. typ. inf.    | szczegóły<br>_techniczne | szczegóły<br>_techniczne  | szczegóły<br>_techniczne  | Ogólne                | szczegóły<br>_techniczne | Ogólne             | Ogólne                    | szczegóły<br>_techniczne | Ogólne                |
| Pasek_opcji        | 2                        | 5                         | 2                         | 5                     | 4                        | 5                  | 4                         | 2                        | 2                     |
| Tło                | 2                        | 5                         | 3                         | 5                     | 4                        | 5                  | 4                         | 2                        | 2                     |
| Podkład_muzyczny   | 2                        | 3                         | 1                         | 5                     | 4                        | 5                  | 4                         | 2                        | 2                     |
| Dźwięki            | 1                        | 4                         | 2                         | 3                     | 4                        | 5                  | 4                         | 2                        | 2                     |
| Głośność_podkładu  | 100                      | 100                       | 100                       | 100                   | 100                      | 100                | 100                       | 100                      | 100                   |
| Głośność_dźwięków  | 100                      | 100                       | 76                        | 100                   | 100                      | 100                | 100                       | 100                      | 38                    |
| Menu_główne        | 2                        | 5                         | 5                         | 5                     | 4                        | 5                  | 4                         | 2                        | 2                     |
| Sekcja_informacje  | 2                        | 5                         | 5                         | 3                     | 4                        | 5                  | 4                         | 2                        | 2                     |
| Sekcja_kolory      | 2                        | 5                         | 2                         | 2                     | 4                        | 5                  | 4                         | 2                        | 2                     |
| Sekcja_galeria     | 2                        | 2                         | 5                         | 2                     | 4                        | 5                  | 4                         | 3                        | 2                     |
| Sekcja_pliki       | 2                        | 2                         | 2                         | 3                     | 4                        | 5                  | 4                         | 2                        | 2                     |
| Satysfakcja        | 4                        | 3                         | 4                         | 4                     | 4                        | 4                  | 4                         | 4                        | 4                     |

Tabela 8. Wartości centroidów po grupowaniu metodą Dattoli

| Klasa         | Procentowy udział | Płeć      | Wiek        | Wykształcenie | Liczba mieszkańców w miejscu zamieszkania | Preferowany typ informacji |
|---------------|-------------------|-----------|-------------|---------------|-------------------------------------------|----------------------------|
| 1             | 33                | mężczyzna | od_25_do_50 | średnie       | poniżej_100.000                           | szczegóły techniczne       |
| 2             | 21                | mężczyzna | od_25_do_50 | wyższe        | od_500.000_do_1.000.000                   | szczegóły techniczne       |
| 3             | 30                | kobieta   | od_25_do_50 | średnie       | poniżej_100.000                           | ogólne                     |
| 4             | 11                | mężczyzna | poniżej_25  | wyższe        | od_500.000_do_1.000.000                   | ogólne                     |
| Izolowana (0) | 5                 | –         | –           | –             | –                                         | –                          |

5. Ocena mechanizmów rejestracji oraz personalizacji, zwłaszcza: satysfakcji z używania określonej konfiguracji systemu oraz możliwości modyfikacji interfejsu i wyglądu aplikacji.

6. Ocena poprawności działania aplikacji, zwłaszcza: poprawność działania poszczególnych opcji systemu, przejść do poszczególnych poziomów menu, poprawność działania mechanizmów rejestracji, personalizacji i logowania, modyfikacji danych osobowych oraz zmian wersji językowych.

Jak widać ocena interfejsu użytkownika była wieloaspektowa. Jest to zgodne ze współczesnymi trendami w rozwoju systemów rekomendacyjnych, które wskazują, aby zamiast pojedynczych ocen, stosowanych np. w przypadku rekomendacji filmów lub książek, wykorzystywać oceny wielokryterialne [Adomavicius 2005], np. rekomendację w restauracjach. Niemniej jednak ostatecznie, np. w celu porównania efektywności poszczególnych interfejsów należy wykorzystać miarę syntetyczną. Miara syntetyczna jest swego rodzaju wypadkową wszystkich branych pod uwagę aspektów i może być stosowana do liniowego porządkowania ocenianych obiektów. W literaturze [Pluta 1986] można znaleźć wiele różnych metod tworzenia miar syntetycznych badających wybrane atrybuty (zmienne, cechy) diagnostyczne. Atrybuty odnoszące się do określonych elementów użyteczności systemów są oczywiście nośnikami różnorodnych informacji, bardzo często od siebie niezależnych [Frokjaer 2000]. W pracy [Kuropka 2006] przedstawiono dwie metody doboru cech diagnostycznych:

- metoda bazująca na znaczeniu danej cechy dla badanego zjawiska,
- metoda bazująca na analizie statystycznej.

W praktyce należy uwzględnić również inne aspekty, a mianowicie możliwości, łatwość i koszt zbierania wartości określonych atrybutów określających użyteczność produktu.

Po wyborze określonych  $m$  cech należy przeprowadzić obserwacje ich wartości dla badanych  $n$  obiektów. Wartości określonej cechy  $j$  ( $j = 1, \dots, m$ ) użyteczności dla okre-

ślonego obiektu  $i$  ( $i = 1, \dots, n$ ) oznaczmy  $u_{i,j}$ , stąd wartości wszystkich  $m$  cech dla obserwowanych  $n$  obiektów można zapisać w macierzy  $\mathbf{U}$ :

$$\mathbf{U} = \begin{bmatrix} u_{1,1} & u_{1,2} & \dots & \dots & u_{1,m} \\ u_{2,1} & u_{2,2} & \dots & \dots & u_{2,m} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{n,1} & u_{n,2} & \dots & \dots & u_{n,m} \end{bmatrix}$$

Mając daną macierz  $\mathbf{U}$ , można określić wartości normalizacyjne, które są stałe w ramach określonego badania:

- $\max_i \{u_{i,j}\}$ : największa wartość cechy  $j$  wśród  $n$  obiektów,
- $\min_i \{u_{i,j}\}$ : najmniejsza wartość cechy  $j$  wśród  $n$  obiektów,
- $\text{nomu}_j$ : określa wartość cechy  $j$ , która jest wzorcowa wśród  $n$  obiektów.

W zależności od kierunku wpływu określonej cechy na użyteczność produktu można wyróżnić następujące ich typy [Kuropka 2006]:

- symulanty, których rosnące wartości wskazują pozytywny wpływ na użyteczność produktu (np. procent zrealizowanych zadań, ocena satysfakcji w rosnącej skali, itp.) mogą być wyrażone wzorem:

$$z_{i,j} = \frac{u_{i,j}}{\max_i \{u_{i,j}\}}, \quad (56)$$

gdzie  $\max_i \{u_{i,j}\} > 0$ ,

- desymulanty, których rosnące wartości wskazują na negatywny wpływ na użyteczność produktu (np. średni czas realizacji zadania, ocena doświadczonego dyskomfortu w rosnącej skali, itp.), mogą być wyrażone wzorem:

$$z_{i,j} = \frac{\min_i \{u_{i,j}\}}{u_{i,j}}, \quad (57)$$

gdzie  $u_{i,j} > 0$ ,

- nominanty, których wzrost odchylenia od pewnej nominalnej wartości ma negatywny wpływ na użyteczność produktu, mogą być wyrażone wzorami:

$$z_{i,j} = \frac{u_{i,j}}{\text{nomu}_j}, \quad (58)$$

gdzie  $u_{i,j} < \text{nomu}_j$ ,

$$z_{i,j} = \frac{\text{nomu}_j}{u_{i,j}}, \quad (59)$$

gdzie  $u_{i,j} < \text{nomu}_j$ ,

Po zdefiniowaniu stymulantu, desymulantu i nominantu można przedstawić definicję zmiennych syntetycznych, które mogą być określone jako suma odpowiednich wartości cech:

$$z'_i = \sum_{j=1}^m z_{i,j} \quad (60)$$

lub jej znormalizowana do przedziału  $[0,1]$  wersja osiągająca wartość 1 dla wzorca może być określona wzorem:

$$z'_i = \frac{1}{m} \sum_{j=1}^m z_{i,j}. \quad (61)$$

Podane rozwiązanie wyznaczania zmiennych syntetycznych jest odpowiednie dla przypadku, w którym wszystkie cechy są mierzalne. Natomiast w przypadku gdy pomiar wartości cech był realizowany przez nadawanie im określonej wartości pewnej skali porządkowej, w pracy [Kuropka 2006] proponuje się, aby traktować takie zmienne jako quasi-mierzalne, zachowujące się przy uporządkowaniu rosnącym jak symulanty.

Proces tworzenia zmiennych syntetycznych może zostać uzupełniony o regulację ważności każdej cechy  $j$ , dla  $j = 1, \dots, m$ , przez nadanie jej wagi  $w_j$  określonej np. na podstawie opinii ekspertów. Jeżeli wagi będą miały wartości z przedziału  $[0, 1]$ , to zagwarantujemy, że zmienna syntetyczna również będzie z tego zakresu. Przy podanym założeniu zmienna syntetyczna będzie wyrażona wzorem:

$$z'_i = \frac{1}{m} \sum_{j=1}^m w_j z_{i,j}. \quad (62)$$

Celem tworzenia tego typu zmiennych syntetycznych jest konstrukcja liniowego uporządkowania obiektów charakteryzującego się pewnym zestawem zróżnicowanych cech. W celu określenia własności dyskryminacyjnej skonstruowanej zmiennej można zdefiniować taki współczynnik zmienności zmiennej  $z$  dla poszczególnych obiektów [Kuropka 2006], aby jego duża wartość świadczyła o zróżnicowaniu cechy syntetycznej. Współczynnik dyskryminacji określony jest w następujący sposób:

$$V_z = \frac{s_z}{\bar{z}}, \quad (63)$$

gdzie:

$$\bar{z} = \frac{1}{n} \sum_{i=1}^n z'_i, \quad (64)$$

$$s_z = \sqrt{\frac{1}{n} \sum_{i=1}^n (z_i - \bar{z})^2}. \quad (65)$$

W badaniu użyteczności interfejsu użytkownika można przytoczyć następujące przykłady cech systemu interakcyjnego:

- symulanty: procent zrealizowanych poprawnie zadań, subiektywna ocena interfejsu dokonywana przez użytkownika,
- desymulanty: liczba popełnianych błędów, procent niezrealizowanych zadań,
- nominanty: czas realizacji zadania, liczba wybranych opcji, czas patrzenia na obszar zawierający szukaną informację.

W przypadku ankiety oceniającej użyteczność interfejsu użytkownika mamy do czynienia jedynie z symulantami. Dodatkowo, poszczególne sześć aspektów można traktować jako jednakowo istotne dla oceny użyteczności interfejsu użytkownika, dlatego do wyznaczenia syntetycznej oceny interfejsu użytkownika stosowano wzór (60). Po jego zastosowaniu dla średnich (po wszystkich użytkownikach) ocen wszystkich sześciu aspektów wartość syntetycznej wartości użyteczności dla ankiet przeprowadzonych po etapie pierwszym wynosi 8,42, a po etapie trzecim 8,72. Można zaobserwować wzrost wartości użyteczności po zastosowaniu rekomendacji kolaboratywnej (CF) w porównaniu z rekomendacją demograficzną (DF) o 0,3 na skali od 0 do 10. Wzrost ten nie jest duży, także i liczba użytkowników biorących udział w eksperymencie nie była wystarczająca, aby osiągnięte wyniki można uznać za statystycznie znaczące, jednak są one ciekawe i zwiększenie użyteczności interfejsu użytkownika (lub przynajmniej brak obniżenia) nastąpił w każdej analizowanej kategorii [Sobecki 2004a].

## 5.7. Podsumowanie

Przedstawiono zastosowanie metody consensusu do rekomendacji interfejsów użytkownika. Opisano architekturę systemu rekomendacji interfejsu użytkownika do webowego systemu informacyjnego. Jako dziedzinę zastosowania tego systemu wybrano promocję samochodu osobowego marki Renault typ Megane II, stąd nazwa systemu *Megane*. W systemie tym rekomendacja interfejsu realizowana jest przez zastosowanie podejścia hybrydowego (HA) polegającego na zastosowaniu filtrowania demograficznego (DF) i filtrowania kolaboratywnego (CF). W implementacji tego systemu DF realizowane jest przez zastosowanie wnioskowania stereotypowego, natomiast CF realizowane jest metodą consensusu.

Omówiono również elementy teorii consensusu. Zdefiniowano podstawowe pojęcia systemu konfliktu, sytuacji konfliktowej oraz consensusu, a także własności consensusu. Przedstawiono modyfikację metody consensusu w rekomendacji interfejsów [Nguyen 2003a], którą przedstawiono w pracy [Sobecki 2006]. Modyfikacja ta polega na wskazaniu sposobu zastosowania metod consensusu do różnych metod filtrowania: DF, CF i CBF oraz w efekcie zastosowanie podejścia hybrydowego. Przedstawiono rów-



nież implementację i badania eksperymentalne efektywności rekomendacji interfejsu zrealizowane z systemem *Megane*. Osiągnięte rezultaty pozwalają stwierdzić, że zastosowanie rekomendacji kolaboratywnej umożliwia zwiększenie użyteczności interfejsu użytkownika w porównaniu z zastosowaniem rekomendacji demograficznej. Wyznaczenie użyteczności interfejsu użytkownika realizowane jest przez wykorzystanie ankiety, w której użytkownik ocenia aspekty użyteczności interfejsu. Przedstawiono również metodę wyznaczenia wartości miary syntetycznej obejmujących różne kryteria oceny.

Wykorzystanie metod consensusu to historycznie najwcześniejsze prace Autora w zakresie rekomendacji interfejsów aplikacji systemów rekomendacyjnych, ich wyniki zostały przedstawione w następujących publikacjach: [Nguyen 2000a], [Sobecki 2001], [Nguyen 2001b], [Sobecki 2001a], [Sobecki 2002a], [Nguyen 2002], [Nguyen 2003a], [Sobecki 2004b] i [Sobecki 2005].

## 6. Adaptacja interfejsu użytkownika z wykorzystaniem metod inspirowanych naturą

W ostatnich latach, oprócz dziedzin, takich jak sztuczna inteligencja (ang. *Artificial Intelligence* – AI) i inteligencja obliczeniowa (ang. *Computational Intelligence* – CI), które starają się naśladować ludzką inteligencję przez wykorzystanie obliczeń symbolicznych lub niesymbolicznych, pojawiają się inne dziedziny wiedzy, które można określić jednym, bardzo pojemnym pojęciem inteligencji bionaśladowczej (ang. *Bio-mimetic Intelligence* – BmI). BmI można zaliczyć do ogólniejszej dziedziny wiedzy – biominetyki [Bar-Cohen 2003], która zajmuje się ogólnie technologiami bionaśladowczymi. BmI, w porównaniu z AI i CI, stara się objąć zespół „zachowań inteligentnych”, z szerszej – biologicznej perspektywy, zarówno z perspektywy mikro i makro, szukając inspiracji na poziomie DNA, komórek, organów i systemów biologicznych, nieskomplikowanych lub bardziej złożonych organizmów, a także całych społeczności tych organizmów. Na podstawie tych idei powstały już bardzo popularne metody, które zostały tu przytoczone w kolejności od poziomu mikro do makro: przetwarzanie DNA (ang. *DNA computing*), przetwarzanie ewolucyjne (ang. *evolutionary computing*), przetwarzanie błonnicze (ang. *membrane computing*), przetwarzanie neuronalne (ang. *neural computing*), sztuczny system immunologiczny (ang. *Artificial Immune System* – AIS), inteligencja rojowa (ang. *swarm intelligence*), kolonia mrówek (ang. *ant colony*), inteligentni agenci (ang. *intelligent agents*), i w końcu sieci społeczne (ang. *social networks*) [Cliff 2003].

W rozdziale zaprezentowano dwie metody BmI: AIS i kolonii mrówek w adaptacji interfejsów użytkownika. Prace prowadzone nad wykorzystaniem tych metod były po części prowadzone w ramach działań w Sieci Doskonałości (ang. *Network of Excellence*) NiSIS (*Nature-inspired Smart Information Systems*). W ich rezultacie powstało kilka prac publikowanych [Sobecki 2007], [Sobecki 2007a] i [Sobecki 2008a] i niepublikowanych referatów wygłoszonych na dwóch sympozjach NiSIS w 2006 r.

Indywidualny wkład autora w zastosowanie AIS do rekomendacji interfejsu użytkownika w systemach webowych obejmuje: określenie tego problemu, adaptację metody rekomendacji filmów [Chen 2004] do rekomendacji interfejsu użytkownika, opracowanie założeń projektowych oraz wybór platformy implementacji do systemu *Reporter* służącego do eksperymentalnej weryfikacji zaproponowanej metod, zaplanowanie i udział w przeprowadzeniu eksperymentu oraz opracowanie wniosków na podstawie

osiągniętych wyników. Przedstawiona metoda zastosowania metafory kolonii mrówek do rekomendacji stanowi w całości indywidualny dorobek autora, z wyłączeniem implementacji oprogramowania służącego do weryfikacji metody na podstawie danych z bazy *MovieLens*.

## 6.1. Metoda AIS w adaptacji interfejsów użytkownika

W obecnym czasie, oprócz zastosowań w adaptacji interfejsów użytkownika metod pochodzących z dziedziny AI, ES lub IR, można spotkać także metody należące do równie szerokiej i rozwijającej się dziedziny BmI. Jedną z takich metod jest AIS, który z powodzeniem był już stosowany w różnych implementacjach systemów rekomendacyjnych [Timmis 2004] i [Garret 2005a]. Na gruncie biologii, jako dziedziny wiedzy, system immunologiczny definiowany jest jako system specjalistycznych komórek i organów, którego celem jest ochrona organizmu przed zewnętrznymi biologicznymi wpływami [Garett 2005a]. AIS jest pewnego rodzaju symulacją ludzkiego systemu immunologicznego (ang. *Human Immune System* – HIS), w którym antygeny atakujące ciało człowieka stymulują system immunologiczny do wytwarzania antyciał [Timmis 2004].

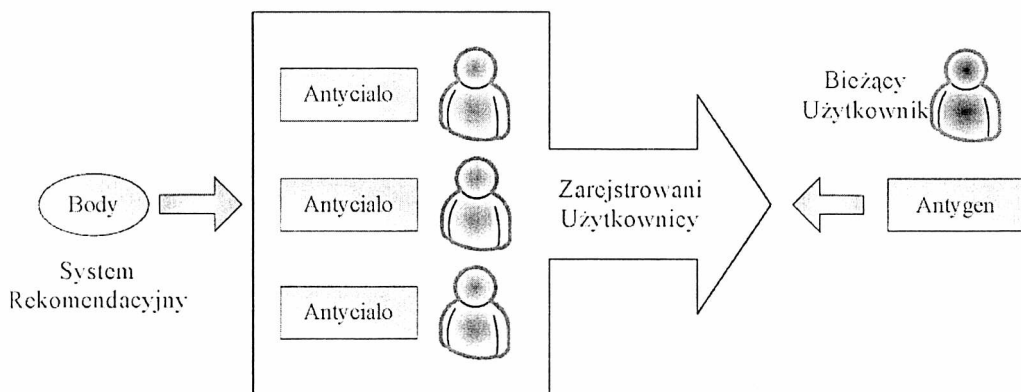
Metoda AIS może zostać użyta na wiele różnych sposobów [Garret 2005a]: jako model wyjaśniający biologiczny system immunologiczny, predykcja lub abstrakcja pewnych systemów immunologicznych. Chociaż metoda AIS jest relatywnie nowa, ma już sporo różnych zastosowań: detekcja zmian (np. spowodowanych przez wirusy komputerowe), detekcja błędów i diagnoza (wykorzystana np. w hardwarowych systemach odpornych na błędy) oraz metoda implementacji selekcji negatywnej.

Metoda AIS w dziedzinie systemów rekomendacyjnych (RS) jest stosowana do tworzenia metod filtrowania kolaboratywnego (CF). W metodzie AIS cały RS identyfikowany jest jako chronione ciało, a zarejestrowani w RS użytkownicy, których preferencje wyrażone np. w postaci dostarczonych ocen, zwani są antyciałami. Użytkownik, któremu mamy dostarczyć rekomendację, zwany jest antygenem. Oczywiście, w ramach HIS, celem systemu immunologicznego jest zwalczenie antygeny przez odpowiednie dobranie antyciał, natomiast w przypadku RS celem jest dostarczenie rekomendacji, która w najlepszy sposób odpowiada użytkownikowi identyfikowanemu jako antygen. Ogólną koncepcję wykorzystania HIS i AIS w RS przedstawiono na rysunku 20.

Celem prezentacji metody AIS jest jej weryfikacja nie tylko w zastosowaniu do rekomendacji konkretnych elementów, ale również jako metoda rekomendacji interfejsu użytkownika.

### 6.1.1. Model AIS zastosowany w RS

Model AIS zastosowany w systemie rekomendacyjnym opiera się na ogólnej koncepcji modelu Farmer–Packard–Perelson (FPP) przedstawionej w pracy [Garret 2005a]. Model ten opiera się na równaniu (66). Mając danych  $N$  antyciał z koncentracjami definiowanymi jako elementy o tych samych cechach łączenia,  $\{x_1, x_2, \dots, x_N\}$  oraz  $n$  anty-



Rys. 20. Ogólna koncepcja wykorzystania HIS-AIS w RS

genów z koncentracjami  $\{y_1, y_2, \dots, y_n\}$ , wtedy zmiana koncentracji  $i$ -tego antyciała wyraża się następującym wzorem:

$$\frac{dx_i}{dt} = c \left[ \sum_{j=1}^N m_{j,i} x_i x_j - k_1 \sum_{j=1}^N m_{i,j} x_i x_j + \sum_{j=1}^n m_{j,i} x_i y_j \right] - k_2 x_i, \quad (66)$$

gdzie:  $c$  – reprezentuje pewną stałą,

$m_{i,j}$  – reprezentuje powinowactwo między koncentracjami odpowiednio o indeksach  $i$  i  $j$ , które nie musi być symetryczne,

$k_1$  – reprezentuje pewną wagę między stymulacją i represją,

$k_2 > 0$  – waga współczynnika śmierci (ang. *death rate*),

$n$  – liczba antygenów.

Zgodnie z modelem koncentracji antyciała, jego wartość rośnie wraz ze wzrostem pokrewieństwa między antyciałami (pierwszy term równania (66), który modeluje stymulację), natomiast maleje wraz ze wzrostem podobieństwa z innymi antyciałami (drugi term, który reprezentuje represję przemnożoną przez pewien współczynnik  $k_1$ ), koncentracja również rośnie wraz ze wzrostem pokrewieństwa między antyciałem i antygenami (trzeci term). Dodatkowo w modelu uwzględniony jest współczynnik śmierci stosowany w przypadku, gdy antyciało nie jest ani dobre ani złe.

Po wyznaczeniu współczynnika koncentracji w AIS następuje sprawdzenie, czy jego wartość jest większa od założonego progu, jeżeli nie, to dane antyciało kasowane jest z AIS i z tymczasowej bazy użytkowników (antyciał) w sposób losowy dobierane jest kolejne antyciało. Proces kończy się w momencie, gdy zgromadzona zostanie odpowiednia liczba antyciał lub gdy w bazie nie ma żadnego przeanalizowanego przez AIS antyciała [Chen 2004].

W zastosowaniu metody AIS w systemie rekomendacyjnym *Reporter* posłużono się schematem częściowo opartym na metodzie zaprezentowanej w pracy [Chen 2004]. Po pierwsze, antygeny i antyciała reprezentowane są przez pewne wektory ocen, a nie we-

który binarne, jak w przypadku modelu FPP. Po drugie, analizowany jest tylko jeden antygen, tzn.  $n = 1$ . Po trzecie, miarą pokrewieństwa nie jest niesymetryczna miara wykorzystująca operację xor między wektorami binarnymi [Garret 2005a], a ważony współczynnik  $\kappa$  Chena przedstawiony w równaniu (19). W końcu, w implementacji AIS w systemie *Reporter* zrezygnowano również z termów represji i śmierci.

### 6.1.2. Architektura systemu *Reporter*

System rekomendacyjny *Reporter* implementujący AIS przeznaczony jest do rekomendacji artykułów z wiadomościami (ang. *news*) z adaptacją interfejsu użytkownika. System *Reporter* wykonuje swoje działania wykorzystując następujące bloki funkcjonalne [Sobecki 2007]:

1. Baza danych przechowująca profile użytkowników zawierające m.in. oceny przeczytanych przez niego artykułów.
2. Rejestracja nowego użytkownika oraz logowanie użytkownika zarejestrowanego oraz oczekiwanie na rekomendację.
3. Algorytm AIS wybiera profile użytkowników o podobnych preferencjach wykorzystując do tego celu ważony współczynnik  $\kappa$  Chena.
4. Po wyborze odpowiednich profili realizowany jest etap predykcji, czyli rekomendowane są odpowiednie artykuły i konfiguracja interfejsu i następuje zaprezentowanie ich bieżącemu użytkownikowi.

Profil użytkownika, oprócz danych rejestracyjnych, zawiera przede wszystkim dane użytkownika w postaci oceny artykułów oraz konfiguracji interfejsu oraz jego oceny. Profile użytkowników reprezentowane są jako krotki. Profil użytkownika inicjowany jest manualnie. Nie wykorzystuje się żadnej techniki uczenia profilu, gdyż dane zbierane są bezpośrednio. W systemie wykorzystuje się sprzężenie zwrotne relewancji w postaci nadawanych ocen artykułów oraz interfejsu użytkownika. Adaptacja profilu następuje po dodaniu każdej nowej oceny, która ma wpływ na kolejne rekomendacje.

W systemie *Reporter* zastosowano kolaboratywną metodę filtrowania informacji (CF). Do określenia dopasowania między profilami wykorzystano algorytm AIS, który bardziej szczegółowo zostanie opisany dalej, natomiast do wyznaczenia predykcji na podstawie określonej przez algorytm AIS wagi zastosowano średnią ważną ocen przedstawioną we wzorze (23):

$$p_{a,j} = \frac{\sum_{i=1}^n w_{a,i} r_{i,j}}{\sum_{i=1}^n w_{a,i}}, \quad (67)$$

gdzie  $w_{a,i}$  reprezentuje wagę zależności między użytkownikiem  $a$  oraz  $i$  wyznaczoną przez algorytm AIS.

Spośród wymienionych bloków funkcjonalnych architektury systemu *Reporter* jedynie blok trzeci, a mianowicie zastosowany algorytm AIS wymaga dokładniejszego

omówienia. Poszczególne kroki algorytmu AIS są modyfikacją algorytmu przedstawionego w punkcie 6.1.1 [Sobecki 2007]:

**Krok 1.** Z bazy danych pobierani są wszyscy użytkownicy, którzy ocenili określoną liczbę wspólnych artykułów z bieżącym użytkownikiem.

**Krok 2.** Wyznaczona jest wartość ważonego współczynnika  $\kappa$  Chena przedstawionego w równaniu (19) między bieżącym użytkownikiem a wszystkimi wyselekcjonowanymi użytkownikami.

**Krok 3.** Wśród wyselekcjonowanych użytkowników wybiera się do dalszej analizy tylko tych użytkowników, dla których ważony współczynnik  $\kappa$  Chena jest większy od określonego progu  $\varepsilon_1$ .

**Krok 4.** Dla wszystkich par wybranych użytkowników wyznaczana jest wartość ważonego współczynnika  $\kappa$  Chena.

**Krok 5.** Do dalszej analizy wybiera się tylko te pary, których wartość ważonego współczynnika  $\kappa$  Chena jest większa od określonego progu  $\varepsilon_2$ .

**Krok 6.** Następnie każdemu wyselekcjonowanemu użytkownikowi nadaje się wagę, która jest stosunkiem liczby par, w których występuje użytkownik i podwojona liczba wszystkich par.

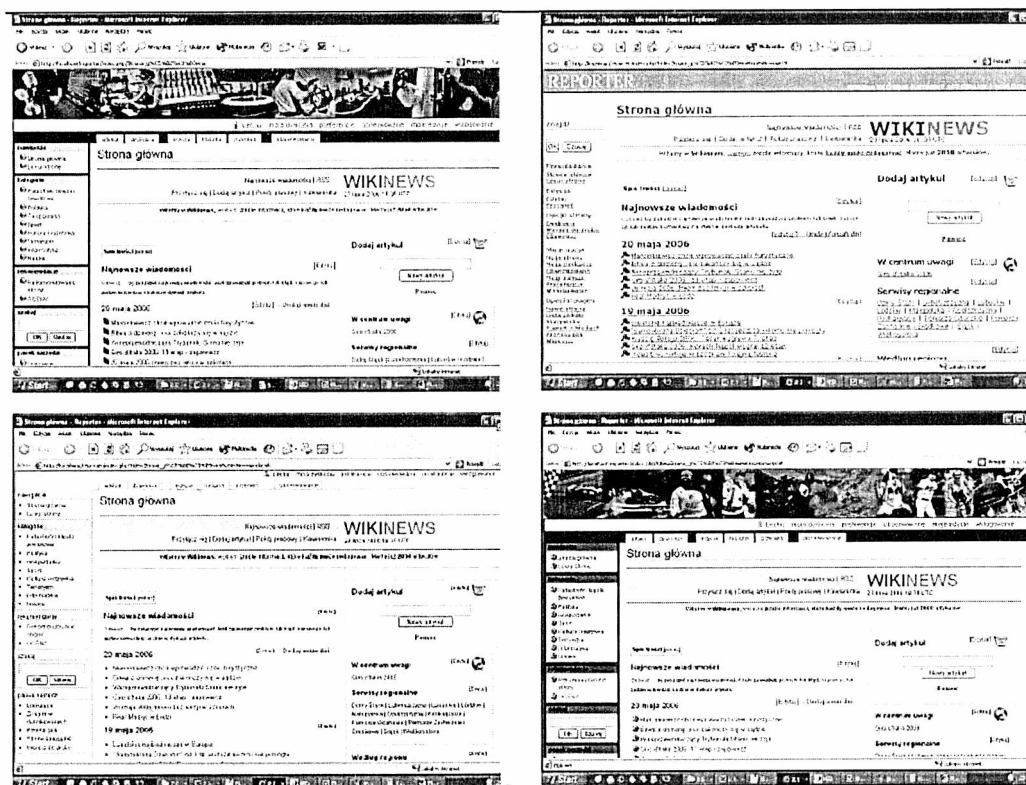
### 6.1.3. Implementacja i weryfikacja

System *Reporter* został zaimplementowany z wykorzystaniem pakietu na licencji free-ware *MetaWiki*. Aby zrealizować funkcjonalność systemu przedstawioną w punkcie 6.1.2, zaimplementowano dodatkowe moduły: dostarczania ocen i rekomendacji. Do bazy z artykułami systemu *Reporter* zostało załadowanych około 2000 pozycji z bazy systemu *WikiNews* [Sobecki 2007]. Jedną z funkcji realizowanych przez system jest rekomendacja interfejsu użytkownika, dlatego zostały zaimplementowane cztery różne kompletne układy graficzne (ang. *layouts*) – patrz rys. 21, które zostały nazwane: multimedia, portal, nonobook i sport. W systemie *Reporter*, tak jak w *MovieLens*, stosowana jest pięciostopniowa skala ocen od 1 – zła, do 5 – bardzo dobra.

Testy systemu *Reporter* zostały przeprowadzone z udziałem grupy 22 użytkowników, przede wszystkim studentów Informatyki i Zarządzania Politechniki Wrocławskiej. Test był realizowany w dwóch etapach. W pierwszym etapie użytkownicy rejestrowali się w systemie, a następnie mogli dowolnie z niego korzystać, czytać i oceniać artykuły. W ramach tego etapu użytkownicy ocenili 967 artykułów. Po analizie zgromadzonych danych oceny dwóch użytkowników zostały odrzucone, w drugim etapie uwzględniono 20 użytkowników i 945 ocenionych przez nich artykułów.

W drugim etapie wzięło udział siedemnastu z wybranych dwudziestu użytkowników. W ramach tego etapu wszyscy użytkownicy zostali poproszeni o ocenę 28 zupełnie nowych artykułów oraz wszystkich czterech układów graficznych interfejsu użytkownika.

Na podstawie przeprowadzonych badań dokonano weryfikację efektywności rekomendacji artykułów i interfejsów użytkownika. Do oceny efektywności wykorzystano tylko jedną z zaprezentowanych w punkcie 4.4.2.1 miarę PA dokładności predykcji (ang.



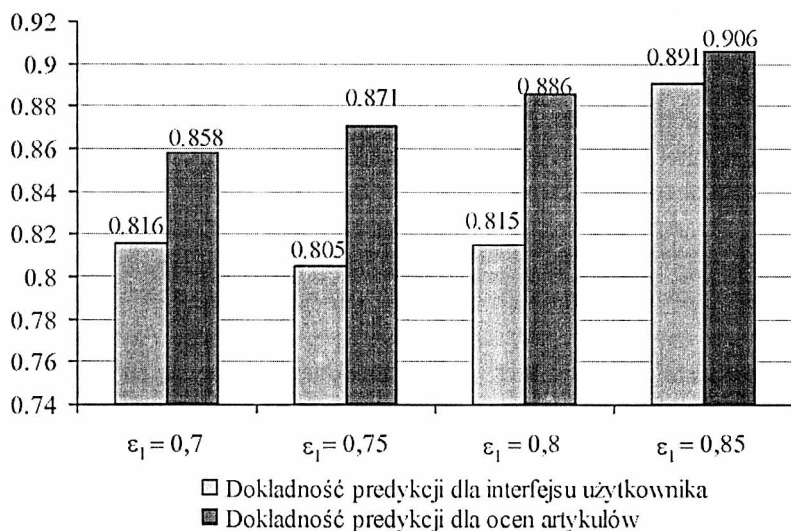
Rys. 21. Zrzuty ekranowe układów graficznych systemu Reporter

*prediction accuracy*) zdefiniowaną przez równanie (42). Wartości tej miary zostały obliczone dla różnych poziomów progu podobieństwa  $\varepsilon_1$  ( $\varepsilon_1 = 0,7$ ;  $\varepsilon_1 = 0,75$ ;  $\varepsilon_1 = 0,8$ ;  $\varepsilon_1 = 0,85$ ), stąd grupy użytkowników podobnych do bieżącego użytkownika zmniejszały się liczebnie wraz ze wzrostem wartości progu  $\varepsilon_1$ , jednak zaobserwowano, że pokrewieństwo między tymi użytkownikami rośnie.

Na podstawie wyników, które przedstawiono na rysunku 22, można zauważyć, że dokładność predykcji artykułów jest dość duża dla wszystkich wartości progu, szczególnie w porównaniu z wynikami badań przeprowadzonymi dla rekomendacji filmów przedstawionymi w tabeli 4. Wartość dokładności predykcji (PA) dla rekomendacji filmów, która wahała się od 81,20% do 82,65%. Wartości dokładności predykcji rekomendacji interfejsu użytkownika jest porównywalna z wartościami dla rekomendacji filmów.

Wyniki dokładności predykcji, które prezentuje rys. 22 pokazują również bardzo ciekawą prawidłowość, taką mianowicie, że dla predykcji artykułów wzrost progu powoduje również wzrost dokładności predykcji. Natomiast prawidłowość ta nie zachodzi w przypadku predykcji interfejsów użytkownika. Powodem takiego stanu może być sposób wyznaczania podobnych użytkowników realizowany na podstawie dostarczonych przez użytkowników ocen artykułów. Jak widać, taki indyktor jest bardzo dobry dla





Rys. 22. Dokładność predykcji dla różnych poziomów progu podobieństwa  $\varepsilon_1$  [Sobecki 2007]

dostarczenia predykcji artykułów, natomiast nieco gorszy dla interfejsów użytkownika. Dodatkowo w przypadku predykcji ocen interfejsów użytkownika można zauważyć, że wzrost progu podobieństwa nie zawsze prowadzi do wzrostu dokładności predykcji. Stąd można zauważyć, że zależność między gustami dotyczącymi treści artykułów i preferencjami co do szaty i układu graficznego interfejsu użytkownika nie da się w pełni potwierdzić. Dlatego podstawowym wnioskiem wynikającym z przeprowadzonych badań nad systemem *Reporter* jest stwierdzenie, że w przypadku stosowania CF podobieństwo użytkowników powinno być wyznaczane na podstawie ocen tych elementów, które są następnie rekomendowane. Drugim, również istotnym wnioskiem jest konstatacja, że AIS można z powodzeniem wykorzystać nie tylko do rekomendacji filmów czy artykułów, ale również do rekomendacji interfejsów użytkownika.

## 6.2. Wykorzystanie metafory kolonii mrówek w adaptacji interfejsów użytkownika

W niniejszym punkcie zostanie przedstawiona m.in. koncepcja wykorzystania ontologii do wymiany informacji o preferencjach użytkownika, a zwłaszcza preferencji dotyczących interfejsu użytkownika wraz z kilkoma metodami inicjalizacji i aktualizacji profilu użytkownika. Profil ten jest następnie wykorzystany przez hybrydową metodę rekomendacji interfejsu użytkownika bazującą na metaforze kolonii mrówek.

Metafora kolonii mrówek jest obecnie dość często wykorzystywana do rozwiązywania różnych kombinatorycznych problemów optymalizacji (ang. *combinatorial optimization problems* – COP), np. problem komiwojażera (ang. *traveling salesman problem*

– TSP) [Colorni 1996] lub kwadratowego problemu przypisania (ang. *the quadratic assignment problem* – QAP) [Deneubourg 1990]. Oczywiście metafora kolonii mrówek zainspirowana została żyjącymi mrówkami, które pomimo że są prawie ślepe, są w stanie znaleźć drogę od gniazda do miejsca żerowania i z powrotem. Mrówki mają tę zdolność dzięki sposobowi komunikacji wewnątrz społeczności mrówek, który wykorzystuje pewną substancję chemiczną zwaną feromonem.

Przedstawiono architekturę webowego systemu rekomendacyjnego z adaptacją interfejsu użytkownika, który wykorzystuje ontologię użytkownika, interfejsu i rekomendowanych elementów. Przedstawiono również wykorzystanie metafory kolonii mrówek do hybrydowej rekomendacji interfejsu użytkownika. Dokonano analizy własności prezentowanej metody oraz jej eksperymentalną weryfikację.

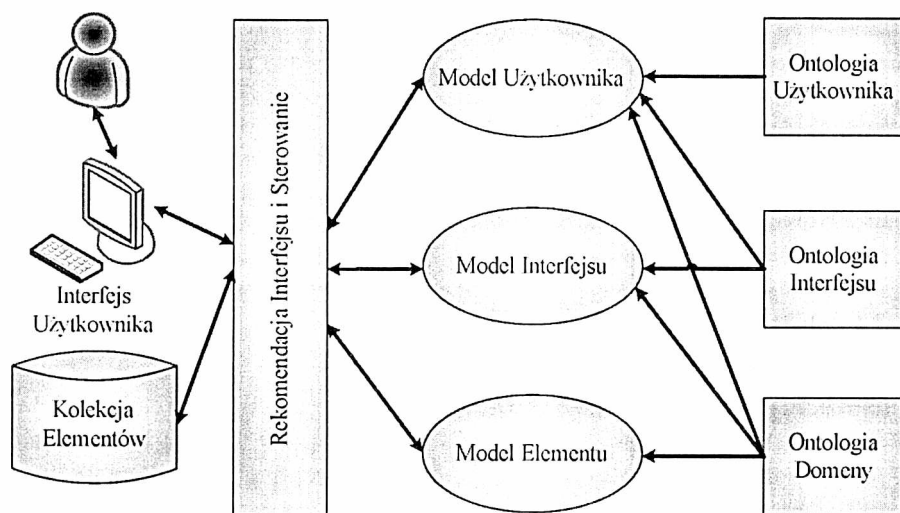
Wyniki badań oparto na w pracach [Sobecki 2007a], [Sobecki 2008a] i [Sobecki 2009].

### 6.2.1. Architektura systemu rekomendacyjnego wykorzystująca ontologię

Centralnym elementem każdego systemu rekomendacyjnego jest model użytkownika [Holt 2002], który zawiera kompozycję przekonań dotyczących danych demograficznych użytkowników, ich preferencji, doświadczeń i wiedzy w określonej dziedzinie rekomendowanych elementów [Razmerita 2003]. Oczywiście każdy webowy system rekomendacyjny jest interakcyjny, dlatego interfejs użytkownika ma duży wpływ na użyteczność całego systemu. Aby zbudować architekturę systemu rekomendacyjnego, należy opracować model użytkownika, model rekomendowanych elementów oraz model interfejsu użytkownika.

Użytkownicy dowolnego systemu webowego najczęściej mają również doświadczenia w wykorzystaniu wielu innych tego typu systemów, dlatego wymiana informacji użytkownika między tymi systemami byłaby bardzo przydatna. Jedną z często stosowanych metod umożliwiających wymianę informacji między różnymi systemami jest zastosowanie ontologii. Ontologia określa podstawowe terminy i kategorie bytów, ich definicje, rodzaje i związki występujące między nimi [Razmerita 2003]. W dziedzinie systemów rekomendacyjnych możemy wyróżnić ontologię użytkownika, ontologię dziedziny związanej z rekomendowanymi elementami oraz ontologię interfejsu użytkownika związaną z układem prezentacji informacji i interakcji. Jednak ze względu na naturę modelu użytkownika, wszystkie te ontologie są wykorzystywane w reprezentacji modelu użytkownika. Ogólny schemat architektury systemu rekomendacyjnego przedstawiono na rysunku 23.

Modelowanie użytkownika jest centralnym problemem wszystkich systemów rekomendacyjnych. Rolą ontologii w modelowaniu jest dostarczenie: słownictwa dla metadanych opisu użytkownika, dziedziny i interfejsu użytkownika, oraz wszelkich relacji między nimi. Model użytkownika, określany również profilem (dokładniej omówiony w punktach 3.3 i 4), obejmuje zawartość (ang. *content*), reprezentację i sposób jego wykorzystania. Zawartość modelu użytkownika obejmuje dwa typy informacji: dane użytkownika i dane użytkownika [Kobsa 2001]. W systemach rekomendacyjnych model użyt-



Rys. 23. Architektura systemu rekomendacyjnego [Sobecki 2008a]

kownika może być reprezentowany na wiele różnych sposobów, np. przez: wektory binarne, wektory cech, krotki, drzewa, drzewa decyzyjne i sieci semantyczne. W prezentowanym w niniejszym punkcie zastosowaniu metafory kolonii mrówek w rekomendacji interfejsów zastosowano reprezentację w postaci krotek oraz grafów.

Rolą ontologii jest oprócz określenia pojęć dostarczenie mechanizmów wnioskowania dla modelu użytkownika, np. określenie grupy podobnych użytkowników, co wymaga określenia funkcji podobieństwa lub odległości w przestrzeni użytkowników oraz metody grupowania, które zostały dokładniej opisane w punktach 3.3.4.2, 4.2.2 i 4.3.1.

Model interfejsu użytkownika w webowym systemie rekomendacyjnym z adaptacją interfejsu użytkownika powinien mieć kilka cech. Powinien być łatwy do określenia i użycia przez projektanta systemu, czyli powinien ułatwiać modelowanie rzeczywistego interfejsu użytkownika w ramach procesu projektowania i implementacji. Zakłada się, że interfejs użytkownika będzie personalizowany, więc jego model powinien uwzględniać łatwość reprezentacji ustawień użytkownika i ich modyfikację dokonywaną on-line. Model powinien uwzględniać łatwą implementację mechanizmów jego rekomendacji. Zastosowanie mechanizmów ontologii powinno umożliwiać wymianę informacji między różnymi modelami interfejsu użytkownika w webowych systemach rekomendacyjnych.

Do reprezentacji modelu interfejsu użytkownika zostały przyjęte elementy teorii grafów. Model interfejsu użytkownika jest więc reprezentowany przez acykliczny graf  $G = (W, E)$ , gdzie  $W$  jest zbiorem węzłów (w literaturze przyjmuje się oznaczenie  $V$ , jednak zostało ono już użyte do oznaczenia zbioru wartości atrybutów), a  $E$  zbiorem krawędzi, gdzie każdy element zbioru  $e(i, j) \in E$ , jest krawędzią łączącą węzły  $w_i$  i  $w_j$ , i ma przyporządkowaną wartość rzeczywistą, która reprezentuje użyteczność dla określone-

go użytkownika  $k$ . Instancja interfejsu użytkownika jest więc ścieżką  $P$  w grafie  $G$ , która zaczyna się w określonym węźle  $s \in W$  i kończy w określonym węźle  $f \in W$ .

Aby zamodelować interfejs użytkownika, należy określić zbiór atrybutów interfejsu  $A^i$  oraz zbiór wartości tych atrybutów  $V^i = \bigcup_{a \in A^i} V_a^i$ . Atrybuty reprezentują różne cechy

interfejsu użytkownika, np. układ graficzny, liczbę kolumn, wybrane tematy zainteresowań, podkład muzyczny i dźwięki. Poszczególne atrybuty i ich wartości są elementami każdego węzła, który zawiera parę: atrybut  $a \in A^i$  ze zbioru atrybutów i wartość atrybutu  $\theta(a)$ , gdzie  $\theta: A^i \rightarrow V^i$  dla  $\forall_{a \in A^i} (\theta(a) \in V_a^i)$ .

W wielu reprezentacjach profili, np. w wektorze cech, zakłada się niezależność poszczególnych atrybutów. Natomiast w przypadku reprezentacji w postaci grafu, takiego założenia się nie przyjmuje, tzn. w rzeczywistym interfejsie, wybór jednej wartości może determinować sensowność kolejnych atrybutów i ich wartości. Dla przykładu, jeżeli wybierzemy wartość atrybutu *liczba\_kolumn* – 2, to określanie wartości zawartości trzeciej kolumny nie ma raczej sensu, gdyż kolumna o tym numerze nie istnieje w tej konkretnej instancji interfejsu użytkownika.

Rozważmy typowy interfejs użytkownika, którego celem jest promocja, reklama i dostarczenie informacji, o określonej marce, firmie lub produkcie (np. modelu samochodu, komputerze czy odbiorniku telewizyjnym). Tego typu system może być odwiedzany nawet przez miliony różnych użytkowników, którzy mają różne preferencje interakcyjne. Prosty, a nawet uproszczony interfejs użytkownika może mieć następujące atrybuty:

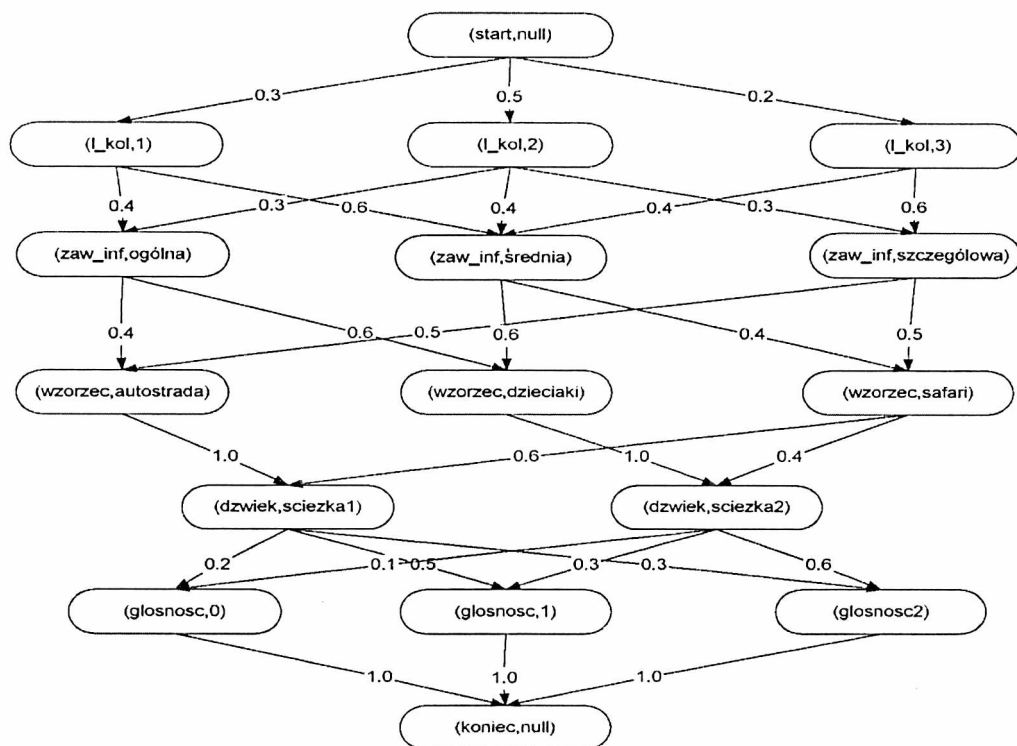
$A = \{\textit{start}, \textit{liczba\_kolumn}, \textit{zawartość\_informacyjna}, \textit{wzorzec}, \textit{dźwięk}, \textit{głośność}, \textit{koniec}\}$

Wartości tych atrybutów przedstawiono w tabeli 9.

Tabela 9. Określenie zbiorów wartości atrybutów

| Lp. | Nazwa zbioru                           | Zbiór wartości                  | Opis                                                                                     |
|-----|----------------------------------------|---------------------------------|------------------------------------------------------------------------------------------|
| 1   | $V_{\textit{start}}$                   | {null}                          | Wartość domyślna                                                                         |
| 2   | $V_{\textit{liczba\_kolumn}}$          | {1, 2, 3}                       | Określa liczbę kolumn tekstu lub elementów interakcyjnych w oknie interfejsu użytkownika |
| 3   | $V_{\textit{zawartość\_informacyjna}}$ | {ogólna, średnia, szczegółowa}  | Określa szczegółowość prezentowanych informacji                                          |
| 4   | $V_{\textit{wzorzec}}$                 | {autostrada, dzieciaki, safari} | Nazwy wzorców interfejsów użytkownika                                                    |
| 5   | $V_{\textit{dźwięk}}$                  | {ścieżka1, ścieżka2}            | Nazwy podkładów muzycznych                                                               |
| 6   | $V_{\textit{głośność}}$                | {0, 1, 2}                       | Poziom dźwięku od najmniejszego do największego                                          |
| 7   | $V_{\textit{koniec}}$                  | {null}                          | Wartość domyślna                                                                         |

Pomimo że przedstawiony tutaj interfejs nie ma raczej zbyt wielu atrybutów, to jednak jego poziom złożoności jest tylko nieznacznie mniejszy niż w przypadku systemu *Megane*. Przynajmniej dwa atrybuty, oprócz atrybutów *start* i *koniec*, mogą być wspólne z wieloma innymi systemami, np. *liczba\_kolumn* i *głośność*, dlatego zastosowanie ontologii może pozwolić na wymianę informacji między różnymi systemami. Przykładowy graf interfejsu użytkownika przedstawiono na rysunek 24.



Rys. 24. Przykładowy graf interfejsu użytkownika [Sobecki 2008a]

Podobieństwo (zwane w terminologii metafory kolonii mrówek widocznością), które określa predykcję relatywnej użyteczności, przypisane zostało do poszczególnych krawędzi. Oznacza to, że wartości te są specyficzne dla każdego użytkownika. Wartości te określają prawdopodobieństwo, stąd suma wszystkich krawędzi wychodzących z każdego węzła musi być równa 1.

## 6.2.2. Metafora kolonii mrówek

Metafora kolonii mrówek znana jest przede wszystkim z jej wykorzystania do rozwiązania optymalizacyjnych problemów kombinatorycznych COP za pomocą algorytmów optymalizacji kolonii mrówek (ang. *ant colony optimization* – ACO), które zostały opracowane w początkach lat 90. ubiegłego stulecia [Colorni 1996]. ACO powstały na bazie obserwacji zachowania się kolonii żywych mrówek.

Mrówki, podobnie jak niektóre inne gatunki takie jak pszczoły, termyty lub niektóre gatunki małp, są zwierzętami społecznymi, żyjącymi w koloniach. Co więcej, role społeczne poszczególnych osobników są bardzo ściśle określone i podporządkowane ogólnemu celowi przetrwania kolonii i opieki nad potomstwem. Jednak w metaforze kolonii mrówek, wykorzystywany jest najczęściej sposób, w jaki mrówki wyszukują najkrótszą drogę do pokarmu. Kolonia mrówek rozwiązuje ten bardzo istotny do istnienia mrowiska problem w sposób bardzo skuteczny i to pomimo że mrówki są prawie ślepe. Mrówki, do komunikowania informacji o pokarmie i jego położeniu, używają specjalnej substancji chemicznej zwanej feromonem [Colorni 1996]. Mrówki, z początku, eksplorują losowo otoczenie mrowiska w poszukiwaniu pokarmu [Blum 2005]. Mrówki poruszając się pozostawiają za sobą feromon, który służy jako znacznik trasy pozwalający zarówno odnaleźć drogę powrotną do gniazda, jak i wskazujący innym mrówkom drogę do pokarmu [Dorigo 1999]. W pracy [Deneubourg 1990] przedstawiono wyniki doświadczenia, które polegało na postawieniu na drodze kolonii mrówek argentyńskich do pokarmu dwóch mostów, górnego i dolnego. Zauważono, że już po upływie 10 minut ponad 90% mrówek wybiera jedną drogę, pomimo że początkowe prawdopodobieństwo wyboru dróg było takie samo. Wy tłumaczeniem takiego zachowania kolonii okazało się składanie i podążanie następców za feromonem tych mrówek, które do gniazda wracały z pokarmem. Wnioskiem z tych badań było opracowanie równania, które uzależnia prawdopodobieństwo wyboru określonej drogi do pokarmu od intensywności pozostawionego śladu feromonu przez inne osobniki. Na jego podstawie przedstawiono w pracy [Colorni 1996] procedurę wyboru optymalnej drogi, która w pracy [Sobecki 2008a] została zaadaptowana do wyboru interfejsu użytkownika.

W punkcie 6.2.1 przedstawiono model interfejsu reprezentowany jest przez acykliczny graf  $G = (W, E)$ , natomiast rekomendacja interfejsu zdefiniowana jest jako znalezienie takiej ścieżki od węzła startowego do węzła końcowego w tym grafie, która ma największą użyteczność dla danego użytkownika. Jak widać, problem znalezienia takiej ścieżki można rozwiązać za pomocą metafory kolonii mrówek, które znajdują ścieżkę wg następującej procedury [Colorni 1996]:

**Krok 1.** Wszystkie mrówki pamiętają odwiedzone już węzły dzięki wykorzystaniu uporządkowanej listy odwiedzonych już węzłów, zwanej *tabu list* (TL).

**Krok 2.** W każdym kroku mrówka wybiera, wykorzystując procedurę wyboru przedstawioną w równaniu (68), węzeł, do którego ma się przemieścić, sprawdzając czy węzeł ten nie jest już obecny na liście TL.

**Krok 3.** Po zakończeniu podróży mrówka składa ślad  $\tau_{i,j}$  na każdej przebytej krawędzi  $e(i, j)$  (gdzie ślad jest analogiem feromonu) i czyści listę TL.

Reguła probabilistyczna wyboru węzła  $j$  przez mrówkę będącą w węźle  $i$  opiera się na następującym równaniu [Colorni 1996]:

$$P(j) = \frac{\tau_{i,j}^\alpha \eta_{i,j}^\beta}{\sum_{l \notin TL} \tau_{i,l}^\alpha \eta_{i,l}^\beta}, \quad (68)$$

gdzie:  $\tau_{i,j}$  – liczba wyrażająca tzw. intensywność śladu na krawędzi  $e(i,j)$ ,  
 $\eta_{i,j}$  – liczba wyrażająca tzw. widoczność węzła  $j$  z węzła  $i$ ,  
 $\alpha, \beta$  – parametry, które sterują relatywny wpływ na intensywność i widoczność, jeżeli wartość tych parametrów jest równa zero, to dany komponent nie jest brany pod uwagę.

W przypadku wykorzystania metafory kolonii mrówek do rekomendacji interfejsu użytkownika, procedura ta musi ulec pewnym modyfikacjom. Po pierwsze, graf interfejsu jest grafem acyklicznym, stąd nie ma możliwości odwiedzenia tego *samego* węzła w grafie więcej niż jeden raz. W konsekwencji nie ma potrzeby używania listy *tabu list* (TL). Zamiast listy TL wykorzystywana jest lista *konfiguracji interfejsu* KI, która zawiera węzły określające ustawienia interfejsu użytkownika. Po drugie, w systemie z rekomendacją interfejsu użytkownika poszczególni użytkownicy są rozróżnialni, nie tak jak w przypadku kolonii mrówek gdzie wszystkie mrówki poszukujące pokarmu są traktowane w ten sam sposób. Ogólna procedura przeszukiwania grafu, czyli wyznaczania konfiguracji interfejsu użytkownika wygląda następująco:

**Krok 1.** Proces rozpoczyna się w węźle *start* oraz zresetowanie listy KI i wstawienie do niego węzła *start*.

**Krok 2.** Na każdym kroku działania procedury wybiera się węzeł, do którego można się przemieścić na podstawie reguły probabilistycznej opisanej w równaniu (68) (dla  $TL = \emptyset$ ) oraz dopisaniu go na końcu listy KI.

**Krok 3.** Krok 2 powtarzany jest aż do czasu, gdy wybranym węzłem jest węzeł *koniec*.

**Krok 4.** Następnie ustalana jest w momencie czasowym  $t_k$  wartość użyteczności  $u_k$  wyznaczonej konfiguracji interfejsu użytkownika  $k$ . Trójka  $(k, u_k, t_k)$  umieszczana jest następnie na wszystkich krawędziach  $e(i,j)$  grafu  $G$ , które łączą węzły umieszczone na liście KI.

Do obliczenia wartości reguły probabilistycznej opisanej w równaniu (68) konieczne jest wyznaczenie wartości widoczności i śladu. Ich wyznaczenie jest jednak bardziej skomplikowane niż w przypadku standardowej procedury. Widoczność  $\eta_{i,j}$  w przypadku rekomendacji interfejsu interpretowana jest jako estymacja użyteczności określonego komponentu interfejsu reprezentowanego przez węzeł  $j$  biorąc pod uwagę, że komponent  $i$  został uprzednio wybrany. Widoczność reprezentuje składową CBF algorytmu, omówioną w punkcie 4.1.2, ponieważ jej wartość jest wyznaczana na podstawie preferencji użytkownika reprezentowanych w jego profilu. Wartość widoczności  $\eta_{i,j}$  wyznaczana jest dla każdej krawędzi wychodzącej węzła  $i$ , w następujący sposób:

1. Jeżeli żadna z wartości atrybutu obecnych w węzłach połączonych krawędziami z węzłem  $i$  nie ma pozytywnej wartości podobieństwa z żadnym atrybutem obecnym w modelu użytkownika, to wartość widoczności dla każdego węzła  $j$  wynosi  $1/n$ , gdzie  $n$  jest liczbą wszystkich węzłów połączonych z węzłem  $i$ .

2. W przeciwnym razie model użytkownika zawiera wartość  $v_a$  dla określonego atrybutu  $a$ , która ma niezerową wartość podobieństwa przynajmniej z jedną wartością tego atrybutu obecną w węźle połączonym krawędzią z węzłem  $i$ . W takim przypadku nale-



ży najpierw obliczyć wartość funkcji podobieństwa między  $v_a$  a każdą wartością tego atrybutu obecną w węzłach połączonych z bieżącym węzłem  $i$ , wtedy wartość widoczności

$$\eta_{i,j} = \frac{\sigma(v_a, v_a^j)}{\sum_{t=1}^n \sigma(v_a, v_a^t)}, \quad (69)$$

gdzie  $n$  jest liczbą wszystkich węzłów połączonych z węzłem  $i$ .

Wartość intensywności obliczana jest, podobnie jak widoczność, dla każdego użytkownika z osobna. Wartość ta reprezentuje składową kolaboratywną metody adaptacji interfejsu bazującą na ocenach użytkowników, którzy wybrali i ocenili użyteczność danej konfiguracji interfejsu. Wartość intensywności  $\tau_{i,j}$  wyznaczana dla każdej krawędzi wychodzącej węzła  $i$  jest równa

$$\tau_{i,j} = \sum_{k=1}^M \rho^k \sigma^k \tau_{i,j}^k, \quad (70)$$

gdzie:  $M$  – liczba użytkowników, którzy w swojej konfiguracji interfejsu wykorzystali wartość z węzła  $j$ ,

$\rho^k$  – parametr określający współczynnik wyparowania dla każdego użytkownika  $k$  oraz ( $0 \leq \rho^k < 1$ ); parametr ten zmniejsza swoją wartość wraz z upływem czasu wyznaczonego przez wartość momentu czasowego  $t_k$ ,

$\sigma^k$  – podobieństwo bieżącego użytkownika z użytkownikiem  $k$  wyznaczone na podstawie wybranej funkcji podobieństwa zaprezentowanej w punkcie 4.2.2,

$\tau_{i,j}^k$  – użyteczność instancji interfejsu dla użytkownika  $k$  wyrażone przez wartość  $u_k$ .

W przypadku gdy  $\tau_{i,j} = 0$  dla każdego  $j$ , wtedy każdemu  $\tau_{i,j}$  nadajemy pewną minimalną wartość  $\varepsilon$ .

### 6.2.3. Badania efektywności i własności wykorzystania kolonii mrówek w rekomendacji interfejsów użytkownika

Najbardziej miarodajnym sposobem weryfikacji określonej metody, zwłaszcza w zakresie nauk stosowanych, jest eksperymentalne badanie jej efektywności. Niestety, nie zawsze jest to w miarę proste, szczególnie ze względu na brak odpowiednio dużej liczby użytkowników. Pomimo, że prezentowane w niniejszej pracy eksperymenty z systemami *Megane* i *Reporter* miały jak dla typowych systemów webowych bardzo niewielką liczbę uczestników, wymagały odpowiednich przygotowań i możliwości ich przeprowadzenia z grupą studencką. Niestety nie zawsze jest to możliwe. Stąd w przypadku wykorzystania metafory mrówek, postanowiono zweryfikować ją w zastosowaniu do rekomendacji filmów. Metoda ta została tak zmodyfikowana, aby umożliwić rekomendację, na podstawie zbioru *MovieLens* (1 milion ocen), którego niektóre cechy zostały już omówione w punkcie 4.3.1. Baza ta zawiera również dane demograficzne 6040 użyt-

kowników, które podają następujące informacje: wiek, płeć, zawód i kod pocztowy, oraz dane dotyczące 3952 filmów, które zawierają: identyfikator filmu, tytuł filmu, data wydania, URL w bazie IMDb, oraz wektor binarny reprezentujący gatunek filmu: nieznan, akcji, przygodowy, animacja, dla dzieci, komedia, kryminał, dokument, dramat, fantasy, film czarno-biały, horror, musical, mystery, romans, sci-fi, thriller, wojenny i western.

Jednak, aby zastosować zaprezentowaną metodę do rekomendacji filmów, czyli predykcji oceny 1, 2, ..., 5, konieczne jest wprowadzenie pewnych modyfikacji do zaproponowanego algorytmu rekomendacji interfejsów użytkownika. Przeszukiwany graf jest bardzo prosty, a mianowicie jest on reprezentowany jako drzewo binarne, które budowane jest na dwa sposoby. W pierwszym, każdy węzeł reprezentuje decyzję czy przyznać filmowi określoną ocenę lub ocenę niższą, czyli w pierwszym węźle podejmowana jest decyzja czy przyznać ocenę 5 lub niższą, analogicznie kolejny węzeł reprezentuje decyzję czy przyznać filmowi ocenę 4 czy niższą, aż do węzłów, gdzie podejmuje się decyzję o przyznaniu oceny 2 lub 1. W drugim sposobie budowy drzewa, każdy węzeł reprezentuje decyzję czy przyznać filmowi ocenę większą lub mniejszą równą od określonej oceny, przy czym rozpoczyna się od oceny 3. W obu sposobach budowy drzewa jego liście reprezentują decyzję, jaką ocenę przyznać danemu filmowi. Do przeszukiwania drogi w tych drzewach wykorzystuje się regułę wyboru przedstawioną w równaniu (67).

Bez względu, czy drzewo zbudowane jest metodą pierwszą czy drugą w każdym węźle, który nie jest liściem, wyznaczenie wartości widoczności  $\eta_{ij}$  i śladu  $\tau_{ij}$ , realizowane jest przez wyznaczenie wartości  $\eta^m$  i  $\tau^m$ , które reprezentują liczby związane z nadaniem określonej oceny  $m$ . Wartość  $\eta^m$  obliczana jest w następujący sposób: tworzymy listę filmów, które zostały ocenione przez danego użytkownika na określoną ocenę  $m$ , a następnie wartość  $\eta^m$  wyznaczamy jako stosunek liczby filmów z listy, które mają przynajmniej jeden gatunek filmu taki sam jak film, dla którego wyznaczamy predykcję oceny do długości listy. Natomiast  $\tau^m$  wyznaczamy jako stosunek liczby podobnych użytkowników, którzy ocenili dany film na ocenę  $m$ , do liczby wszystkich podobnych użytkowników, którzy ocenili dany film. Mając wartości  $\eta^m$  i  $\tau^m$ , dla ocen  $m = 1, 2, \dots, 5$  można wyznaczyć wartości widoczności  $\eta_{ij}$  i śladu  $\tau_{ij}$  jako sumy odpowiednio wartości  $\eta^m$  i  $\tau^m$ , dla takich ocen  $m$ , które występują w węźle  $j$ .

Dla zweryfikowania niniejszej metody przeprowadzono badania z wykorzystaniem zbioru 100 tysięcy ocen MovieLens nadanych przez 943 użytkowników i dotyczących 1682 filmów [GrupLens 2006], który jest skróconą wersją zbioru z 1 milionem ocen. W niniejszych badaniach została użyta procedura parametryczna, która pozwala między innymi na: wybór tworzenia drzewa pierwszą lub drugą metodą, określenie wartości  $\alpha$  i  $\beta$ , oraz określenie liczności sąsiedztwa podobnych użytkowników. Otrzymane rezultaty prezentuje tabela 10.

Wartości te są całkiem dobre, porównując z rezultatami osiągniętymi dla metody CF z wykorzystaniem różnych funkcji podobieństwa i najprostszej metody predykcji, które zostały zaprezentowane w punkcie 4.4.3, z jednoczesnym poprawieniem parametru po-

Tabela 10. Rezultaty badań eksperymentalnych nad zastosowania metafory kolonii mrówek do rekomendacji filmów

| Sposób tworzenia drzewa | $\alpha$ | $\beta$ | Maksymalna liczba sąsiadów | Dokładność predykcji |
|-------------------------|----------|---------|----------------------------|----------------------|
| 1                       | 1        | 1       | 10                         | 74,81%               |
| 1                       | 2        | 1       | 10                         | 74,45%               |
| 1                       | 1        | 2       | 10                         | 74,88%               |
| 1                       | 1        | 1       | 100                        | 79,37%               |
| 1                       | 2        | 1       | 100                        | 79,42%               |
| 1                       | 1        | 2       | 100                        | 79,04%               |
| 2                       | 1        | 1       | 10                         | 75,95%               |
| 2                       | 2        | 1       | 10                         | 75,60%               |
| 2                       | 1        | 2       | 10                         | 76,09%               |
| 2                       | 1        | 1       | 100                        | 81,00%               |
| 2                       | 2        | 1       | 100                        | 80,07%               |
| 2                       | 1        | 2       | 100                        | 80,97%               |

krycia, które wynosi 100% we wszystkich przypadkach, czyli dla każdego użytkownika i każdego filmu jesteśmy w stanie dostarczyć wystarczająco dobrej predykcji.

Zaproponowana tutaj architektura webowego systemu rekomendacyjnego z adaptacją interfejsu użytkownika ma kilka pozytywnych cech, które mogą zostać docenione przez każdego projektanta systemów rekomendacyjnych.

Po pierwsze, jak to było postulowane na początku, architektura ta jest z jednej strony prosta, a z drugiej bardzo łatwa do zaimplementowania. Reprezentacja interfejsu użytkownika w postaci grafu może być łatwo zaimplementowana w postaci dokumentu XML [Sobecki 2003]. Zastosowanie XML pozwala na prostą wymianę danych między różnymi platformami systemowymi i sprzętowymi, tak jak np. w implementacji systemu *Megane*. Głównym problemem na tym etapie jest wystarczająco powszechna akceptacja ontologii, a zwłaszcza ontologii interfejsu użytkownika. Używanie i implementowanie różnego typu standardów jest obecnie nie tylko dobrą praktyką ale nawet koniecznością, choć wdrażanie nowych standardów zwykle wiąże się z wieloma problemami. Jednak nawet w przypadku, gdy nie istnieją ogólnie przyjęte ontologie, można oprzeć się na ontologiach popularnych systemów, których użycie odnotowane jest w historii przeglądarki internetowej, a pewne parametry, w tym interfejsu użytkownika, obecne w pliku *cookies*. Można również wykorzystywać pewne ustawienia przeglądarki, takie jak wielkość tekstu czy zawartość i rozmieszczenie pasków narzędziowych.

Drugą zaletą zastosowania metafory kolonii mrówek jest integracja podejścia hybrydowego w jednej metodzie. W wielu implementacjach [Sobecki 2005] każda z metod filtrowania DF, CF i CBT została zaimplementowana jako osobny element, który był uruchamiany w zależności od etapu działania systemu bądź dostępności odpowiednich

danych. Także zastosowanie kombinacji tych metod może nieść ze sobą pewne problemy z powodu trudności z porównywalnością rezultatów poszczególnych metod.

Trzecią, bardzo ważną zaletą proponowanej architektury, jest możliwość jej wykorzystania bez względu na etap działania systemu. Tak jak wspomniano w punkcie 4.1.2 jedną z wad CF jest niewielka liczba ocen nowych elementów. W przypadku wykorzystania metafory kolonii mrówek probabilistyczna reguła selekcji pozwala na rekomendację zupełnie nowych elementów.

W końcu, po czwarte, proponowana architektura jest niezależna od dziedziny zastosowania i może być implementowana na wielu różnych platformach systemowo-sprzętowych, od stacji roboczych do telefonów przenośnych wyposażonych w przeglądarki webowe.

## 6.3. Podsumowanie

Przedstawiono dwie metody rekomendacji interfejsów użytkownika w webowych systemach adaptacyjnych inspirowane naturą. W pierwszej części zaprezentowano aplikację metody sztucznego systemu immunologicznego AIS, a w drugiej zastosowanie metafory kolonii mrówek.

W literaturze znane są przykłady zastosowania AIS do rekomendacji różnych elementów np. filmów [Chen 2004]. W prezentowanym systemie *Reporter* rekomendowane są zarówno wiadomości, jak i interfejs użytkownika. Przeprowadzone badania potwierdziły wysoką efektywność rekomendacji wiadomości mierzoną dokładnością predykcji. Dokładność ta zwiększała się wraz ze wzrostem progu podobieństwa doboru sąsiedztwa (czyli podobnych użytkowników), czyli im bardziej podobnych użytkowników bierzemy pod uwagę do wyznaczenia predykcji oceny, tym bardziej dokładne rekomendacje artykułów możemy dostarczyć.

Nowością w systemie *Reporter* jest natomiast zastosowanie AIS do rekomendacji interfejsów użytkownika. Uzyskane w eksperymencie dokładności predykcji interfejsu, chociaż mniejsze niż w przypadku rekomendacji artykułów, są jednak bardzo zadowalające. W rekomendacji interfejsów użytkownika nie stwierdzono jednak podobnej zależności jak w przypadku rekomendacji artykułów tzn. wzrostu dokładności wraz ze wzrostem podobieństwa między użytkownikami mierzoną tak jak w przypadku rekomendacji artykułów na podstawie ocen artykułów. Stąd można wyprowadzić dosyć ważny wniosek z przeprowadzonych badań, a mianowicie że w CF następuje wzrost dokładności rekomendacji wraz ze wzrostem podobieństwa użytkowników bazującego na ocenach rekomendowanych elementów. Jednocześnie można stwierdzić, że zależność ta nie musi być prawdziwa w przypadku, gdy podobieństwo między użytkownikami wyznaczane jest na podstawie ocen pewnych elementów, a rekomendacje dotyczą zupełnie innych rzeczy.

Drugą zaprezentowaną metodą inspirowaną naturą było wykorzystanie metafory kolonii mrówek. W ramach prac nad wykorzystaniem tej metafory opracowano metodę zastosowania ontologii do modelowania użytkownika, interfejsu użytkownika i reko-

mendowanego elementu. Kolejnym krokiem było opracowanie oryginalnej metody rekomendacji interfejsu użytkownika reprezentowanych w postaci grafu z wykorzystaniem metafory kolonii mrówek. Metoda ta w odróżnieniu od wielu innych metod umożliwia uwzględnienie występowania zależności między różnymi atrybutami opisującymi interfejs użytkownika. Kolejną zaletą niniejszej metody jest zintegrowanie dwóch podstawowych metod filtrowania CF i CBF w pojedynczej metodzie przez odpowiednie zdefiniowanie parametrów widzialności i śladu w analizowanym grafie. Metoda ta została zweryfikowana z wykorzystaniem danych symulacyjnych dotyczących rekomendacji filmów z bazy MovieLens. Wyniki tych prac zostały przedstawione w [Sobecki 2007a], [Sobecki 2008a] i [Sobecki 2009].

## 7. Wykorzystanie klasyfikacji przybliżonej do wyznaczenia optymalnego profilu użytkownika

Na etapie projektowania systemu rekomendacyjnego bardzo szybko podejmowana jest decyzja dotycząca zawartości profilu użytkownika. Projektanci decydują, jakie informacje dotyczące użytkowników będą zbierane, a następnie wykorzystywane, m.in. do rekomendacji w określonym systemie. Decyzje te nie zawsze muszą być najlepsze. Po pierwsze, może nastąpić sytuacja, w której pewne informacje będą zbierane zupełnie bez potrzeby. Po drugie zaś, może się okazać, że występuje brak określonych danych [Nguyen 2005].

Niniejszy rozdział prezentuje metodę, która wykorzystuje klasyfikację przybliżoną [Pawlak 1999] do wyznaczenia zestawu optymalnych atrybutów w profilu użytkownika. Za podstawę analizy przyjęto dane zebrane podczas badań systemu *Megane*, stąd zarówno zawartość i metoda reprezentacji profilu użytkownika, jak i funkcje odległości oraz metoda grupowania zostały zaczerpnięte z tego projektu.

### 7.1. Koncepcja i podstawowe pojęcia klasyfikacji przybliżonej

Pojęcie klasyfikacji przybliżonej zostało wprowadzone przez Pawlaka [Pawlak 1999] i związane jest z pojęciem systemu informacyjnego [Pawlak 1983] określonego w definicji 12.

#### Definicja 12

System informacyjny jest czwórką uporządkowaną  $S = (X, A, V, \rho)$ , gdzie  $X$  jest zbiorem obiektów,  $A$  jest zbiorem atrybutów opisujących obiekty,  $V$  jest zbiorem wartości atrybutów i  $\rho: X \times A \rightarrow V$  jest funkcją informacyjną taką, że  $\rho(x, a) \in V_a$ , gdzie dla każdego  $x \in X$ ,  $a \in A$  oraz  $V = \bigcup_{a \in A} V_a$ .

System informacyjny może być przedstawiony w tabeli, gdzie kolumny reprezentują atrybuty, a wiersze poszczególne wartości funkcji informacyjnej dla poszczególnych obiektów. W tabeli 7 podano przykład takiego systemu, gdzie każda wartość  $\rho(x, a)$  funkcji  $\rho$  jest umieszczona na przecięciu rzędu  $x$  z kolumną  $a$ . Obiekty reprezentowane przez

wartości atrybutów można klasyfikować w klasy (zwane również zbiorami opisywalnymi) na podstawie identycznej wartości krotek.

W pracach Pawlaka została zaprezentowana metodologia pozwalająca wyznaczyć minimalny zbiór atrybutów, zwany *reduktem*, który generuje taką samą klasyfikację jak cały zbiór atrybutów. Metodologia ta jest bardzo przydatna, gdyż jej rezultatem może być wyznaczenie niewielkiego zbioru atrybutów, który generuje taką samą klasyfikację jak pierwotny rozbudowany ich zbiór.

Podstawową rolę w określeniu klasyfikacji przybliżonej odgrywa pojęcie zbioru przybliżonego [Pawlak 1983]. Niech  $S = (X, A, V, \rho)$  będzie systemem informacyjnym oraz niech  $Y \subseteq X$ , wtedy największy (w sensie zawierania) zbiór opisywalny w systemie  $S$  zwany jest *dolnym przybliżeniem zbioru*  $Y$  w systemie  $S$  i oznaczany  $\underline{S}Y$ . Analogicznie, najmniejszy (w sensie zawierania) opisywalny zbiór w systemie  $S$  zwany jest *górnym przybliżeniem zbioru*  $Y$  w systemie  $S$  i oznaczany  $\overline{S}Y$ . Analogicznie można oznaczyć dolne i górne przybliżenie zbioru  $Y$  w redukcje systemu  $S$  do zbioru atrybutów  $B \subset A$  jako:  $\underline{S}_BY$  i  $\overline{S}_BY$ .

Mając zdefiniowaną pewną klasyfikację  $C$  zbioru  $X$ , w pracy [Pawlak 1983] wprowadza się pojęcie klasyfikacji przybliżonej w następujący sposób. Niech klasyfikacja  $C$  będzie generowana przez pewien zbiór atrybutów  $B \subset A$ , wtedy *przybliżenie* klasyfikacji  $C$  określone jest przez pewien zbiór  $B' \subset B$ , w ten sposób, że klasyfikacja  $C'$  generowana przez zbiór  $B'$  różni się nieznacznie od klasyfikacji  $C$ . Różnica między klasyfikacjami może być przedstawiona jako pewna miara określona przez równanie (71), która powinna być mniejsza od założonego progu  $\varepsilon$

$$\mu_B(Y) - \mu_{B \setminus B'}(Y) \leq \varepsilon \quad (71)$$

dla każdego  $Y \subseteq X$ , gdzie

$$\mu_L(Y) = \frac{\text{card}(\underline{S}_LY)}{\text{card}(\overline{S}_LY)}, \quad (72)$$

dla pewnego  $L \subseteq A$ .

W pracy [Nguyen 2005] przedstawiona została inna koncepcja konstrukcji zbioru przybliżonego opartego na pojęciu odległości zdefiniowanej w przestrzeni klasyfikacji generowanych przez poszczególne zbiory atrybutów. Problem określenia zbioru atrybutów, który generuje najbardziej podobną klasyfikację należy do dziedziny klasyfikacji przybliżonej, jednak nie był według naszej wiedzy wcześniej rozważany.

Przedstawienie problemu wymaga określenia pewnych dodatkowych pojęć: podziału, relacji zawierania się podziałów, podziału generowanego przez zbiór atrybutów oraz dwóch typów odległości między podziałami.

### Definicja 13

Podziałem zbioru  $X$  nazywamy niepuste i rozłączne klasy, podzbiory zbioru  $X$ , których suma jest równa  $X$ .



**Definicja 14**

Niech  $\pi(X)$  oznacza zbiór wszystkich podziałów zbioru  $X$  oraz niech  $P, Q \in \pi(X)$ , wtedy produkt podziałów  $P$  i  $Q$ , oznaczony jako  $P \cap Q$  zdefiniowany jest jako:  $P \cap Q = \{p \cap q: p \in P, q \in Q \text{ i } p \cap q \neq \emptyset\}$ .

Łatwo pokazać, że tak zdefiniowany produkt  $P \cap Q$  jest również podziałem zbioru  $X$ .

**Definicja 15**

Niech  $P, Q \in \pi(X)$ , wtedy mówimy, że podział  $P$  zawiera się w podziale  $Q$ , co oznaczamy  $P \subseteq Q$ , wtedy i tylko wtedy, gdy każda klasa  $P$  zawiera się w klasie podziału  $Q$ .

**Definicja 16**

Podziałem określonym przez atrybut  $a \in A$  nazywamy taki podział  $P_a \in \pi(X)$  zbioru  $X$ , którego elementy składają się z obiektów  $x \in X$  mających identyczną wartość atrybutu  $a$ .

**Definicja 17**

Podziałem określonym przez zbiór atrybutów  $B \subseteq A$  nazywamy taki podział  $P_B \in \pi(X)$  zbioru  $X$ , którego elementy składają się z obiektów  $x \in X$  mających wartości takie same jak dla poszczególnych atrybutów  $a \in B$ .

Nie jest trudno pokazać, że  $P_B = \bigcap_{b \in B} P_b$ .

Można zdefiniować funkcję odległości między podziałami opartą na reprezentacji podziału w postaci macierzy. Niech  $P, Q \in \pi(U)$  oraz  $M(P) = [p_{i,j}]_{n \times n}$  będzie symetryczną macierzą kwadratową  $n \times n$ , gdzie  $n = \text{card}(X)$ , taką, że:

$$p_{i,j} = \begin{cases} 0 & \text{dla } x_i, x_j \text{ w różnych klasach } P \\ 1 & \text{dla } x_i, x_j \text{ w tych samych klasach } P \end{cases}$$

oraz macierz  $M(Q) = [q_{i,j}]_{n \times n}$  będzie określona w analogiczny sposób, wtedy odległość między macierzami  $P$  i  $Q$  może być zdefiniowana w następujący sposób:

$$\delta(P, Q) = \frac{1}{2} \sum_{i,j=1}^n |p_{i,j} - q_{i,j}|. \quad (73)$$

Tak zdefiniowana odległość jest metryką. Można łatwo pokazać, że złożoność obliczeniowa wyznaczenia odległości  $\delta$  wynosi  $O(n^2)$  [Musiał 1989].

### Przykład

Weźmy pod uwagę system informacyjny, który przedstawiono w tabeli 7 i podziały, które generują odpowiednio atrybuty  $P$  – przez płeć i  $Q$  – przez wykształcenie, wtedy  $P = \{\{pk, student, dyk, jazzman, pawelg\}, \{podymak, alusia, allisza, jakecz\}\}$ ,  $Q = \{\{pk, student, podymak, alusia\}, \{allisza, dyk, jakecz, jazzman, pawelg\}\}$ . Odległość między podziałami  $P$  i  $Q$  wynosi 20.

## 7.2. Klasyfikacja przybliżona profili użytkownika

W systemie *Megane* zaprezentowanym w rozdziale 5 przedstawiono wykorzystanie metod consensusu. W systemie tym użytkownicy klasyfikowani są na podstawie danych demograficznych do odpowiednich klas, do których następnie przyporządkowane są odpowiednie rekomendacje interfejsu. Pojawia się tutaj pewne pytanie: czy atrybuty demograficzne służące do tej klasyfikacji są rzeczywiście odpowiednie do rekomendacji interfejsów? Innymi słowy, należy odpowiedzieć na pytanie, na ile podział generowany przez atrybuty demograficzne odpowiada podziałowi generowanemu przez atrybuty konfiguracji interfejsu.

Przyjmijmy odpowiednie oznaczenia wykorzystywane w systemie *Megane* przytoczone w punkcie 5.4, który opisuje profil użytkownika. Możemy teraz określić dwa podziały, podział  $P_C$ , gdzie  $C = \{wiek, płeć, wykształcenie, liczba\_mieszkańców\_w\_miejscu\_zamieszkania, preferowany\_typ\_informacji\} \subseteq A$  zawiera atrybuty demograficzne i preferencji, które służyły do klasyfikacji użytkowników opisanej w punkcie 5.5, oraz podział  $Q$ , który jest podziałem bazującym na atrybutach interfejsu użytkownika  $B = \{pasek\_opcji, tło, podkład\_muzyczny, dźwięki, głośność\_podkładu, głośność\_dźwięków, menu\_główne, sekcja\_informacje, sekcja\_kolory, sekcja\_galeria, sekcja\_pliki\} \subseteq A$ .

Mając zdefiniowane podziały  $P_C$  i  $Q$ , można rozróżnić dwa przypadki. Pierwszy, najbardziej korzystny, gdy podziały są sobie równe, wtedy mówimy, że podział  $P_C$  jest **optymalny i w pełni podatny na adaptację**. W drugim odległości między podziałami są różne od zera, wtedy wraz ze wzrostem odległości zbiór atrybutów pogarsza swoją jakość, przy czym w tej sytuacji można również wyznaczyć dwa przypadki. Pierwszy, gdy  $P_C$  lub dowolny z podziałów generowany przez podzbiór  $C$  ma odległość do podziału  $Q$  mniejszą od założonego progu, wtedy mówimy, że podział  $P_C$  **jest podatny na adaptację**, natomiast zbliżenie do rozwiązania optymalnego można próbować uzyskać przez eliminację pewnych atrybutów, które mogą być zbędne. W drugim, gdy  $P_C$  i każdy z podziałów generowany przez podzbiór  $C$  ma odległość do podziału  $Q$  większą od założonego progu, wtedy mówimy, że podział  $P_C$  **nie jest podatny na adaptację**, gdyż w zbiorze może brakować pewnych atrybutów i profil powinien zostać o nie uzupełniony.

W przypadku gdy  $P_C$  nie jest optymalny, jednak zakładając, że podział  $P_C$  jest podatny na adaptację, szukamy takiego podzbioru zbioru  $C$ , który będzie **wystarczająco zbliżony do rozwiązania optymalnego**, tzn. odległość między podziałem generowa-

nym przez ten podzbiór a podziałem  $Q$  jest mniejsza od założonego progu. Mówimy również, że podział  $P_C$  **jest bliższy do rozwiązania optymalnego** niż podział  $P_{C'}$ , jeżeli odległość między podziałem generowanym przez podzbiór  $C$  a podziałem  $Q$  jest mniejsza niż odległość między podziałem generowanym przez podzbiór  $C'$  a podziałem  $Q$ . Poszukiwanie podzbioru zbioru  $C$ , wystarczająco zbliżonego do rozwiązania optymalnego, jest problemem, który należy do dziedziny klasyfikacji przybliżonej (ang. *rough classification* – RC). W pracy [Nguyen 2006] przedstawiono jego rozwiązanie, dekomponując go na trzy mniejsze problemy klasyfikacji przybliżonej (RC), które można zdefiniować w następujący sposób:

### Problem RC-1

*Dla danej partycji  $Q$  zbioru użytkowników  $U$  należy określić zbiór  $D \subseteq C$  zbioru danych użytkownika, taki, że  $P_D \subseteq Q$  i  $\delta(P_D, Q)$  jest minimalna.*

W problemie RC-1 zbiór  $D$  tworzy taki podział, który jest minimalnie bardziej dokładny niż podział  $Q$ . Taki podział, reprezentuje klasyfikację, która gwarantuje, że żadnych dwóch użytkowników, którzy spersonalizowali swoje interfejsy w różny sposób (podział  $Q$ ) nie należy do tej samej klasy. W przypadku, gdy istnieje rozwiązanie problemu RC-1, mamy do czynienia z *dokładną* klasyfikacją, w przeciwnym razie należy spróbować rozwiązać problem RC-3.

### Problem RC-2

*Dla danej partycji  $Q$  zbioru użytkowników  $U$  należy określić zbiór  $D \subseteq C$  zbioru danych użytkownika, taki że  $Q \subseteq P_D$  i  $\delta(P_D, Q)$  jest minimalna.*

Rozwiązanie tego problemu reprezentuje przypadek, gdy podział generowany przez  $D$  jest minimalnie mniej dokładny niż podział  $Q$ . Taki podział, reprezentuje klasyfikację, która, niestety, nie gwarantuje, że żadnych dwóch użytkowników, którzy spersonalizowali swoje interfejsy w różny sposób (podział  $Q$ ), nie należy do tej samej klasy, jednak liczba takich przypadków powinna być niewielka. Rozwiązanie tego problemu nie daje optymalnego rozwiązania, jednak pozwala na znalezienie takiego zbioru atrybutów, który jest mniejszy niż zbiór atrybutów będący rozwiązaniem problemu RC-1.

### Problem RC-3

*Dla danej partycji  $Q$  zbioru użytkowników  $U$  należy określić zbiór  $D \subseteq C$  zbioru danych użytkownika, taki że  $\delta(P_D, Q)$  jest minimalna.*

Rozwiązanie tego problemu może być konieczne, w przypadku gdy nie można znaleźć rozwiązania problemu RC-1 i problemu RC-2. W większości praktycznych zastosowań, należy nawet pominąć próby rozwiązania problemu RC-1 i problemu RC-2, i przejść bezpośrednio do rozwiązywania problemu RC-3, którego rozwiązaniem jest otrzymanie zbioru atrybutów  $D$  będącego suboptymalnym zbiorem w świetle przybliżonej klasyfikacji  $Q$ .

### Przykład

Weźmy pod uwagę system informacyjny przedstawiony w tabeli 7, gdzie zbiór użytkowników  $U = \{pk, student, podymak, alusia, allisza, dyk, jakecz, jazzman, pawelg\}$ ,

podziałem  $P_C$  będzie podział generowany przez wartości zbioru atrybutów  $C$ , gdzie  $C = \{\text{wiek}, \text{płeć}, \text{wykształcenie}, \text{liczba\_mieszkańców\_w\_miejscu\_zamieszkania}, \text{preferowany\_typ\_informacji}\} \subseteq A$  zawiera atrybuty demograficzne i preferencji, które służyły do klasyfikacji użytkowników opisanej w punkcie 5.5, oraz podział  $Q$ , który jest podziałem bazującym na atrybutach interfejsu użytkownika  $B = \{\text{pasek\_opcji}, \text{tło}, \text{podkład\_muzyczny}, \text{dźwięki}, \text{głośność\_podkładu}, \text{głośność\_dźwięków}, \text{menu\_główne}, \text{sekcja\_informacje}, \text{sekcja\_kolory}, \text{sekcja\_galeria}, \text{sekcja\_pliki}\} \subseteq A$ . Stąd:

$$\begin{aligned} P_{\{\text{płeć}\}} &= \{\{pk, \text{student}, \text{dyk}, \text{jazzman}, \text{pawelg}\}, \{\text{podymak}, \text{alusia}, \text{allisza}, \text{jakecz}\}\}, \\ P_{\{\text{wiek}\}} &= \{\{pk, \text{student}, \text{podymak}\}, \{\text{allisza}, \text{dyk}, \text{jakecz}, \text{jazzman}, \text{pawelg}\}, \{\text{alusia}\}\}, \\ P_{\{\text{wykształcenie}\}} &= \{\{pk, \text{student}, \text{podymak}, \text{alusia}\}, \{\text{allisza}, \text{dyk}, \text{jakecz}, \text{jazzman}, \text{pawelg}\}\}, \\ P_{\{\text{l.mieszk.w\_msc.zam.}\}} &= \{pk, \text{student}, \text{allisza}, \text{dyk}, \text{pawelg}\}, \{\text{podymak}, \text{jakecz}, \text{jazzman}, \text{alusia}\}\}, \\ P_{\{\text{pref.typ\_inf.}\}} &= \{pk, \text{podymak}, \text{alusia}, \text{dyk}\}, \{\text{student}, \text{allisza}, \text{jakecz}, \text{jazzman}, \text{pawelg}\}\} \end{aligned}$$

stąd

$$\begin{aligned} P_C &= \{\{pk\}, \{\text{student}\}, \{\text{podymak}\}, \{\text{alusia}\}, \{\text{allisza}\}, \{\text{dyk}\}, \{\text{jakecz}\}, \{\text{jazzman}\}, \{\text{pawelg}\}\} \\ Q &= P_B = \{\{pk\}, \{\text{student}\}, \{\text{podymak}, \text{allisza}\}, \{\text{alusia}\}, \{\text{dyk}\}, \{\text{jakecz}\}, \{\text{jazzman}\}, \{\text{pawelg}\}\} \end{aligned}$$

Mamy więc sytuację, gdzie  $P_C \subseteq Q$ , stąd problem RC-1 powinien mieć swoje rozwiązanie, a tym rozwiązaniem są zbiory  $\{\text{płeć}, \text{wykształcenie}, \text{liczba\_mieszkańców\_w\_miejscu\_zamieszkania}, \text{preferowany\_typ\_informacji}\}$  oraz  $\{\text{wiek}, \text{płeć}, \text{liczba\_mieszkańców\_w\_miejscu\_zamieszkania}, \text{preferowany\_typ\_informacji}\}$ , gdyż w tym przypadku atrybut  $\text{wykształcenie}$  zależny jest od atrybutu  $\text{wiek}$  [Pawlak 1983], odległość zaś  $\delta = 1$ . Rozwiązaniem problemu RC-2 jest  $D = \{\text{płeć}\}$ , które jest niestety dalekie od ideału. Rozwiązaniem problemu RC-3 jest rozwiązanie problemu RC-1.

### 7.3. Rozwiązania problemów klasyfikacji przybliżonej

Aby można było praktycznie wykorzystać rozwiązania problemów RC-1, RC-2 i RC-3, należy przedstawić algorytmy ich obliczania. Jednak zanim zostaną one zaprezentowane należy zaprezentować algorytm wyznaczania produktu dwóch podziałów  $P = \{p_1, p_2, \dots, p_k\}$  i  $R = \{r_1, r_2, \dots, r_l\}$  zbioru  $U$ ,  $S = P \cap R$ . W podanym algorytmie wykorzystano własność symetryczności macierzy reprezentującej podział.

#### Algorytm wyznaczania produktu podziałów

Dane: Podziały  $P$  i  $R$  przedstawione w postaci macierzy,  $n = \text{card}(U)$   
 Rezultat: Produkt  $S = P \cap R$  przedstawiony w postaci macierzy

START

OD  $i:=1$  DO  $n$  WYKONAJ

OD  $j:=1$  DO  $i$  WYKONAJ

JEŻELI  $p_{i,j} = r_{i,j} = 1$ , TO  $s_{i,j} := 1$ ;  $s_{j,i} := 1$ ;

W PRZECIWNYM PRZYPADKU  $s_{i,j} := 0$ ;  $s_{j,i} := 0$ ;

KONIEC.

### 7.3.1. Rozwiązanie problemu RC-1

Łatwo zauważyć [Nguyen 2005], że jeżeli  $P_D \subseteq Q$ , to  $P_C \subseteq Q$ , gdyż  $P_C \subseteq P_D$  ( $P_C = P_D \cap P_{CD}$ ), stąd jeżeli  $P_C \not\subseteq Q$  to nie istnieje rozwiązanie problemu RC-1. Jednak w przeciwnym razie (tzn. gdy  $P_C \subseteq Q$ ) rozwiązanie tego problemu istnieje i może nim być dowolny zbiór  $D \subseteq C$ , pod warunkiem, że wartość funkcji odległości  $\delta(P_D, Q)$  jest minimalna. Niniejsze rozważania są oparte na twierdzeniu 8 [Nguyen 2006].

#### Twierdzenie 8

Dla  $P, Q, R \in \pi(U)$ , jeżeli  $P \subseteq Q$ , to  $\delta(P, Q) \leq \delta(P \cap R, Q)$ .

#### Dowód

Łatwo zauważyć, że jeżeli podziały  $P$  i  $Q$  są przedstawione w postaci macierzy i jeżeli  $P \subseteq Q$ , to dla każdego elementu macierzy  $P, Q, P \cap R$ : jeżeli  $p_{i,j} = 1$ , to  $q_{i,j} = 1$ , oraz jeżeli  $p \cap r_{i,j} = 1$ , to  $p_{i,j} = 1$  (czyli macierz  $Q$  ma przynajmniej tyle lub więcej jedynek co macierz  $P$ ), stąd z definicji odległości  $\delta$  wynika, że odległość  $\delta(P \cap R, Q)$  może być większa lub co najmniej równa  $\delta(P, Q)$ . ■

Na podstawie twierdzenia 8 można wykazać, że jeżeli eliminacja jakiegoś atrybutu  $a \in D \subseteq C$  powoduje, że ciągle spełniona jest zależność  $P_{D-\{a\}} \subseteq Q$ , to tak otrzymany podział ma odległość mniejszą do podziału  $Q$  niż podział przed eliminacją tego atrybutu ( $\delta(P_{D-a}, Q) \leq \delta(P_D, Q)$ ).

#### Algorytm RC-1 [Nguyen 2006]

Dane: Podziały  $P_a$ , dla każdego  $a \in C$  oraz  $Q$  zbioru  $U$ .

Wynik:  $D \subseteq C$  taki, że  $P_D \subseteq Q$  i  $\delta(P_D, Q)$  jest prawie minimalna (wynik przybliżony)

START

Wyznacz  $P_C$ ;

JEŻELI  $P_C \not\subseteq Q$ , TO  $D := \emptyset$  I IDŹ NA KONIEC.

$D := C$ ;

DLA KAŻDEGO  $a \in C$

JEŻELI  $P_{D \setminus \{a\}} \subseteq Q$ , TO  $D := D \setminus \{a\}$ ;

KONIEC.

Rozwiązanie, które znajduje ten algorytm jest suboptymalne, a złożoność obliczeniowa tego algorytmu wynosi  $O(m^2n^2)$ , gdzie  $m$  jest liczbą atrybutów w zbiorze  $C$ , a  $n$  liczbą elementów zbioru  $U$ , czyli użytkowników, których profile analizujemy.

### 7.3.2. Rozwiązanie problemu RC-2

Zanim zostanie przedstawione rozwiązanie problemu RC-2, zostanie przedstawione twierdzenie 9, które prezentuje zależności odległości dla wzajemnie zawierających się podziałów. Łatwo zauważyć, że zależność  $Q \subseteq P_D$  zachodzi wtedy i tylko wtedy, gdy zachodzi  $Q \subseteq P_a$  dla każdego  $a \in D$ . Do znalezienia rozwiązania Problemu RC-2 wykorzystane zostaną własności zaprezentowane w twierdzeniu 9.

#### Twierdzenie 9

Dla  $P_a, P_b, Q \in \pi(U)$ :

a) jeżeli  $Q \subseteq P_a \subseteq P_b$ , to  $\delta(P_a, Q) \leq \delta(P_b, Q)$ ,

b) jeżeli  $Q \subseteq P_a$  i  $Q \subseteq P_b$ , to  $\delta(P_{\{a\} \cup \{b\}}, Q) \leq \delta(P_a, Q)$  i  $\delta(P_{\{a\} \cup \{b\}}, Q) \leq \delta(P_b, Q)$ .

Dowód zostanie przeprowadzony na podstawie pracy [Nguyen 2006].

#### Dowód

a) Łatwo zauważyć, że jeżeli podziały  $P_a, P_b$  i  $Q$  przedstawione są w postaci macierzy i odległości  $\delta(P_a, Q) = k$ , to  $Q \subseteq P_a \subseteq P_b$  dla każdego elementu macierzy  $Q \subseteq P_a$ : jeżeli  $q_{ij} = 1$ , to  $p_{ij}^a = 1$  oraz dalej jeżeli  $p_{ij}^a = 1$ , to  $p_{ij}^b = 1$ , stąd  $\delta(P_a, Q) = k + l$ , gdzie  $l \geq 0$ , stąd  $\delta(P_a, Q) \leq \delta(P_b, Q)$ . ■

b) Łatwo zauważyć, że jeżeli zachodzi  $Q \subseteq P_a$  i  $Q \subseteq P_b$ , to zachodzą również  $Q \subseteq P_{\{a\} \cup \{b\}} \subseteq P_a$  oraz  $Q \subseteq P_{\{a\} \cup \{b\}} \subseteq P_b$ , stąd z a) mamy  $\delta(P_{\{a\} \cup \{b\}}, Q) \leq \delta(P_a, Q)$  i  $\delta(P_{\{a\} \cup \{b\}}, Q) \leq \delta(P_b, Q)$ . ■

Łatwo zauważyć, że zależność  $Q \subseteq P_D$  zachodzi wtedy i tylko wtedy, gdy zachodzi  $Q \subseteq P_a$  dla każdego  $a \in D$ . Do znalezienia rozwiązania problemu RC-2 wykorzystane zostaną własności zaprezentowane w twierdzeniu 9, które mówią, że zbiór  $D$  może zostać zminimalizowany przez eliminację atrybutów zależnych [Pawlak 1983], tzn. jeżeli  $a, b \in D$  są takie, że  $P_a \subseteq P_b$ , to atrybut  $b$  jako zależny od  $a$  może być usunięty z podziału.

#### Algorytm RC-2 [Nguyen 2006]

Dane: Podziały  $P_a$ , dla każdego  $a \in C$  oraz  $Q$  zbioru  $U$ .

Wynik: Zbiór  $D \subseteq C$  taki, że  $Q \subseteq P_D$  i  $\delta(P_D, Q)$  jest prawie minimalny (wynik przybliżony)

START

$D := \emptyset$ ;

DLA KAŻDEGO  $a \in A$

JEŻELI  $Q \subseteq P_a$ , TO  $D := D \cup \{a\}$ ;

DLA KAŻDEGO  $a, b \in A$  I  $a \neq b$

JEŻELI  $P_a \subseteq P_b$ , TO  $D := D \setminus \{b\}$ ;

KONIEC.

Złożoność obliczeniowa tego algorytmu wynosi tak jak algorytmu RC-1,  $O(m^2n^2)$ .

### 7.3.3. Rozwiązanie problemu RC-3

Problem RC-3 jest NP-zupełny [Musiał 1989]. W pracy [Nguyen 2006] zostały zaprezentowane trzy algorytmy heurystyczne dla rozwiązania tego problemu. W niniejszej pracy zostanie przytoczony algorytm oznaczony w pracy [Nguyen 2006] numerem 3, który został zweryfikowany eksperymentalnie oraz algorytm genetyczny [Boudjeloud 2005].

Idea algorytmu polega na rozpoczęciu tworzenia zbioru  $D$  od elementu zbioru  $C$ , którego podział jest najbliższy do szukanego podziału  $Q$  i powtarzaniu tego kroku, aż do spowodowania zwiększenia odległości.

#### Algorytm RC-3

Dane: Podziały  $P_a$ , dla każdego  $a \in C$  oraz  $Q$  zbioru  $U$ .

Wynik: Zbiór  $D \subseteq C$  taki, że  $\delta(P_D, Q)$  jest prawie minimalny (wynik przybliżony)

START

Wybierz  $a \in C$  taki, że  $\delta(P_a, Q) = \min_{b \in C} \delta(P_b, Q)$ ;

$D = \{a\}$ ;

$E := D$ ;

DOPÓKI  $D \neq E$  WYKONAJ

JEŻELI istnieje  $c \in C \setminus D$  taki, że

$$\delta(P_{D \cup \{c\}}, Q) = \min_{b \in C \setminus D} \delta(P_{D \cup \{b\}}, Q) \text{ i } \delta(P_{D \cup \{c\}}, Q) < \delta(P_D, Q),$$

TO  $D := D \cup \{c\}$ ;

KONIEC.

Złożoność obliczeniowa tego algorytmu wynosi, jak algorytmów RC-1 i RC-2,  $O(m^2n^2)$ .

Bardzo często do rozwiązywania problemów optymalizacyjnych wykorzystywane są algorytmy genetyczne (ang. *genetic algorithms* – GA) [Holland 1975], stąd można je również zastosować do znalezienia rozwiązania Problemu RC-3. Algorytmy genetyczne należą do szerszej rodziny algorytmów ewolucyjnych. Algorytmy ewolucyjne, opisywane na poziomie abstrakcyjnym, implementują zasady wynikające z praw biologicznych [Zieliński 2000]: poszukiwanie rozwiązania z wykorzystaniem procesów ewolucyjnych w populacji rozwiązań; dziedziczenie informacji przez wybrane rozwiązania w kolejnych generacjach populacji rozwiązań.

Algorytmy genetyczne rozwiązujące problemy optymalizacyjne dotyczące zmiennych zakodowanych w postaci ciągów binarnych, których odpowiednie wartości należy zoptymalizować nazywa się binarnymi (ang. *binary GA*) [Haupt 2004]. Natomiast w przypadku wykorzystania ciągłych wartości zmiennych mamy do czynienia z ciągłymi algorytmami genetycznymi.



Binarny GA realizowany jest w następujących krokach [Haupt 2004]:

**Krok 1.** Określenie funkcji przystosowania i optymalizowanych zmiennych.

**Krok 2.** Określenie parametrów GA.

**Krok 3.** Losowe wygenerowanie początkowej populacji.

**Krok 4.** Dekodowanie chromosomów.

**Krok 5.** Wyznaczenie wartości funkcji przystosowania dla wszystkich chromosomów.

**Krok 6.** Wyznaczenie chromosomów – kandydatów na rodziców następnej generacji.

**Krok 7.** Krzyżowanie.

**Krok 8.** Mutacja.

**Krok 9.** Sprawdzenie warunku spełnienia określonego kryterium.

**Krok 10.** Jeżeli warunek jest spełniony wtedy algorytm kończy swoje działanie, w przeciwnym przypadku przejdź do wykonania kroku 4.

Binarny algorytm genetyczny bardzo prosto można wykorzystać do rozwiązywania np. problemu RC-3. Poszukiwane rozwiązanie łatwo jest przedstawić w postaci łańcucha binarnego, reprezentującego poszczególne atrybuty ze zbioru  $C$ , o długości równej liczbie analizowanych atrybutów ze zbioru  $C$  –  $\text{card}(C)$ . Optymalizowanymi zmiennymi są obecność (1) lub nieobecność (0) każdego atrybutu ze zbioru  $C$  w poszukiwanym zbiorze  $D$ . Wszystkie te zmienne tworzą chromosom. Ponieważ reprezentacja problemu jest binarna, nie istnieje potrzeba dekodowania chromosomów. Funkcją przystosowania będzie wartość funkcji  $\delta(D, Q)$ . Natomiast kryterium końca może być liczba generacji. W kroku 2 należy określić dwa pozostałe parametry GA, a mianowicie prawdopodobieństwo krzyżowania i prawdopodobieństwo mutacji. Parametry te zostały ustalone odpowiednio na 0,5 oraz 0,01.

## 7.4. Przykład rozwiązania problemów RC-1, RC-2 i RC-3

Przedstawiono rozwiązanie problemów RC-1, RC-2 i RC-3 dla systemu informacyjnego, który przedstawiono w tabeli 7; zbiór  $C$  został określony tak, jak w punkcie 7.2,  $C = \{\text{wiek}, \text{płeć}, \text{wykształcenie}, \text{liczba\_mieszkańców\_w\_miejscu\_zamieszkania}, \text{preferowany\_typ\_informacji}\} \subseteq A$ , natomiast zbiór  $Q$  został określony na podstawie grupowania Dattoli, zrealizowanego na podstawie wartości zbioru atrybutów  $B = \{\text{pasek\_opcji}, \text{tło}, \text{podkład\_muzyczny}, \text{dźwięki}, \text{głośność\_podkładu}, \text{głośność\_dźwięków}, \text{menu\_główne}, \text{sekcja\_informacje}, \text{sekcja\_kolory}, \text{sekcja\_galeria}, \text{sekcja\_pliki}\} \subseteq A$ , które określają parametry konfiguracji interfejsu użytkownika. W rezultacie otrzymano podział  $Q = \{\{\text{podymak}, \text{alusia}, \text{dyk}\}, \{\text{student}, \text{jakecz}, \text{pawelg}\}, \{\text{pk}\}, \{\text{allisza}, \text{jazzman}\}\}$ . Wartość podziału  $Q$  jest inna, niż w przykładzie zaprezentowanym w punkcie 7.2, gdyż podział ten powstał na podstawie grupowania profili użytkownika za pomocą algorytmu Dattoli [Dattola 1968].

Dla rozwiązania problemu RC-1 należy znaleźć  $D \subseteq C$  takie, że  $P_D \subset Q$  i  $\delta(P_D, Q)$  jest minimalna. Zgodnie z algorytmem RC-1 najpierw sprawdza się, czy zależność  $P_C \subseteq Q$  jest prawdziwa. Następnie weryfikuje się, czy  $P_{C \setminus \{a\}} \subseteq Q$  dla wszystkich  $a \in C$ ,

co jest prawdziwe jedynie dla atrybutu *wiek*, czyli rozwiązaniem jest zbiór atrybutów  $D = C \setminus \{wiek\}$ . Konsekwencją tego rozwiązania jest możliwość usunięcia atrybutu  $\{wiek\}$  z profilu, przy jednoczesnej możliwości zachowania selektywności w odniesieniu do zgrupowanych profili użytkownika.

Rozwiązanie problemu RC-2 oznacza określenie takiego  $D \subseteq C$ , dla którego  $Q \subseteq P_D$  i  $\delta(P_D, Q)$  jest minimalna. Zgodnie z algorytmem RC-2 najpierw znajduje się atrybut  $a \in C$  taki, że  $Q \subseteq P_{\{a\}}$ . W analizowanym systemie nie można znaleźć żadnego takiego atrybutu, stąd, w tym przypadku, nie istnieje rozwiązanie problemu RC-2. Jak widać, w rzeczywistej sytuacji znalezienie rozwiązania tego problemu może być trudne. Pozostaje jedynie próba rozwiązania problemu RC-3.

W celu rozwiązania problemu RC-3 należy wyznaczyć takie  $D \subseteq C$ , w którym  $\delta(P_D, Q)$  jest minimalna. W algorytmie RC-3 najpierw określa się  $\delta(P_a, Q)$  dla wszystkich  $a \in C$ . Odległość  $(P_a, Q)$  jest minimalna dla  $a = \text{preferowany\_typ\_informacji}$ . W drugim kroku określana jest odległość  $\delta(P_{\{a\} \cup \{\text{preferowany\_typ\_informacji}\}}, Q)$  dla wszystkich  $a \in C \setminus \{\text{preferowany\_typ\_informacji}\}$ , która jest minimalna dla  $a = \text{wiek}$  i mniejsza niż  $\delta(P_{\{\text{preferowany\_typ\_informacji}\}}, Q)$ . Kolejny krok algorytmu, niestety, nie poprawia rozwiązania, stąd wynikiem działania algorytmu RC-3 jest  $D = \{\text{preferowany\_typ\_informacji}, \text{wiek}\}$ , dla którego  $\delta(P_D, Q) = 10$ . Nie jest to rozwiązanie optymalne, gdyż rozwiązanie problemu RC-1  $D = C \setminus \{wiek\}$  wyznacza odległość o wartości 7.

Zaprezentowane rozwiązania pokazują, że dobre rezultaty można osiągnąć dla zbiorów atrybutów zawierających atrybut *wiek* oraz takich, które go nie zawierają. Jest to spowodowane faktem, że istnieje w zbiorze  $C$  atrybut *wykształcenie*, od którego atrybut *wiek* zależy [Pawlak 83].

Złożoność obliczeniowa algorytmów RC-1, RC-2 i RC-3, tak jak o tym wspomniano odpowiednio w punktach 7.3.1, 7.3.2 i 7.3.3 wynosi  $O(m^2n^2)$ , gdzie  $m$  jest liczbą atrybutów w zbiorze  $C$ , a  $n$  jest liczbą elementów zbioru  $U$ , czyli użytkowników, których profile analizujemy. Dla analizowanego przykładu wartości te są raczej niewielkie, jednak nawet w przypadku wielu tysięcy użytkowników algorytmy te powinny działać efektywnie, zwłaszcza, że operacje te są wykonywane w tle i nie wpływają bezpośrednio na działanie systemu.

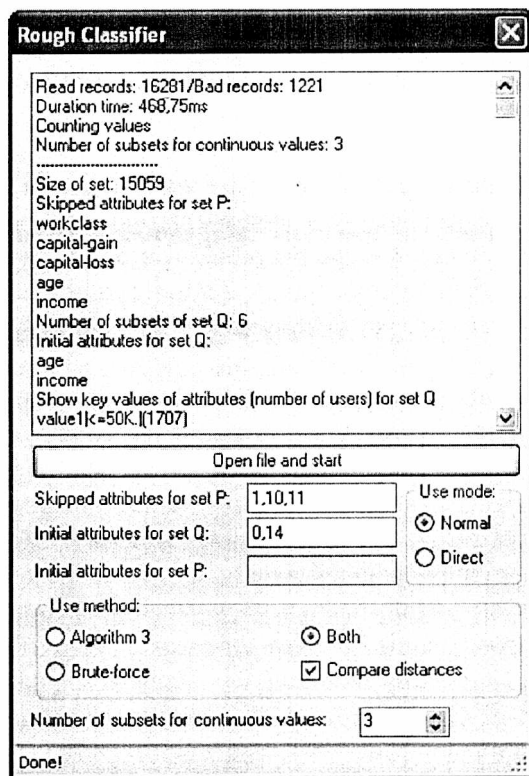
Sprawdzono również algorytm genetyczny do rozwiązania problemu RC-3. Do przeprowadzenia eksperymentu wykorzystano może nie najbardziej efektywną pod względem obliczeniowym implementację algorytmu genetycznego w języku Prolog [Nilson 1990], jednak umożliwiła ona sprawne przystosowanie jej na potrzeby niniejszego problemu. Implementacja ta została opracowana przez Alberta Ali Salah z Uniwersytetu w Bogazici w Turcji. Do tej implementacji zostały wprowadzone konieczne poprawki dotyczące wczytywania systemu informacyjnego, wyznaczania podziałów oraz wyznaczania odległości między podziałami. Algorytm ten pozwala modyfikować następujące parametry: rozmiar populacji (który musi być wyrażony liczbą parzystą), rozmiar chromosomu, liczbę generacji, prawdopodobieństwo krzyżowania pomnożone przez 10 000 i prawdopodobieństwo mutacji również pomnożone przez 10 000. W niniejszym przykładzie parametry te zostały ustawione odpowiednio na: 10, 5, 10, 5000 i 100. Ponie-

waż algorytmy genetyczne nie są deterministyczne, więc przy różnych uruchomieniach można uzyskać różne rezultaty, i tak w pierwszych czterech uruchomieniach uzyskano cztery różne, ale jednakowo optymalne rezultaty, dla których odległość między podziałami wynosi 7:  $C$ ,  $C \setminus \{\text{wiek}\}$ ,  $C \setminus \{\text{wykształcenie}\}$ ,  $C \setminus \{\text{wiek}, \text{wykształcenie}\}$ . Podobnie dobre rezultaty można osiągnąć, redukując liczbę generacji, choć częściej zdarzają się rozwiązania, które w rezultacie generują podziały mniej optymalne.

## 7.5. Badania eksperymentalne klasyfikacji przybliżonej

Przykład zaprezentowany w punkcie 7.4 wykorzystuje próbkę o nieznaczonej wielkości i może służyć jedynie jako demonstracja idei rozwiązania. W celu zweryfikowania efektywności działania przybliżonego algorytmu RC-3 została zaimplementowana aplikacja *Rough Classifier* w środowisku Visual Studio 2005 .NET, w języku C# [Nguyen 2006]. Okno aplikacji przedstawia rysunek 25.

Aplikacja *Rough Classifier* realizuje trzy podstawowe funkcje: wczytanie zadanego zbioru danych, znajdowanie atrybutów, które generują podział  $P$  najbliższy do zadanego podziału  $Q$  za pomocą przybliżonego algorytmu RC-3, oraz znajdowanie tego podziału



Rys. 25. Przykładowe okno aplikacji *Rough Classifier* [Nguyen 2006]

za pomocą algorytmu przeszukiwania całej przestrzeni rozwiązań (ang. *brute-force*), która sprawdza wszystkie sensowne kombinacje atrybutów.

W eksperymencie został wykorzystany zbiór danych *Adult*, który jest bardzo popularny w dziedzinie eksploracji danych. Dane te zostały wyekstrahowane z bazy danych biura spisu ludności (ang. *census bureau*), która udostępniona została pod adresem <http://www.census.gov/ftp/pub/DES/www/welcome.html> przez Ronny Kohavi i Barry Becker z Data Mining and Visualization Silicon Graphics. Zbiór danych, po usunięciu danych o nieznanym znaczeniu, został podzielony na dwie części: *data1*, zawierający dane o 30162 osobach oraz *data2*, zwany zbiorem testowym i zawierający dane 15059 osób. Zbiór *Adult* [Adult 1998] ma 15 atrybutów: 0 – wiek (ang. *age*), 1 – kategoria zatrudnienia (ang. *work-class*), 2 – wykształcenie (ang. *education*), 3 – identyfikator wykształcenia (ang. *education-num*), 4 – stan cywilny (ang. *marital-status*), 5 – zawód (ang. *occupation*), 6 – stan rodzinny (ang. *relationship*), 7 – rasa (ang. *race*), 8 – płeć (ang. *sex*), 9 – przychód z kapitału (ang. *capital-gain*), 10 – strata na kapitale (ang. *capital-loss*), 11 – różnica na kapitale (ang. *fnlwgt*), 12 – liczba godzin pracy w tygodniu (ang. *hours-per-week*), 13 – kraj pochodzenia (ang. *native-country*), 14 – przychód (ang. *income*). Ponieważ niektóre wartości atrybutów są ciągłe, więc na potrzeby działania aplikacji *Rough Classifier* należało ograniczyć zmienność ich wartości do określonej liczby przedziałów.

Dzięki aplikacji *Rough Classifier* możemy rozwiązać Problem RC-3 za pomocą dwóch algorytmów: przybliżonego algorytmu RC-3 i algorytmu przeszukiwania całej przestrzeni rozwiązań; jako parametry możemy podać: zbiór danych, atrybuty, których wartości generują podział wzorcowy  $Q$  oraz zbiór atrybutów  $C$ , z których należy wybrać podzbiór  $D$  minimalizujący wartość odległości  $\delta(P_D, Q)$ . Co prawda zbiór *Adult* został przygotowany do weryfikacji różnych metod predykcji czy dana osoba zarabia więcej niż 50K (tysięcy dolarów) rocznie, czy nie, to dzięki aplikacji *Rough Classifier* można zweryfikować różne korelacje między zbiorami atrybutów. W celu weryfikacji Algorytmu RC-3 pod względem optymalności rozwiązania oraz czasu obliczeń została przeprowadzona seria eksperymentów z wykorzystaniem zbiorów danych *data1* i *data2* dla różnych zbiorów generujących podział wzorcowy  $Q$  oraz zbiorów  $C$ . Aplikacja została uruchomiona na komputerze klasy PC z dwoma procesorami AMD Athlon 2200+ i 1 GB RAM, pod systemem Microsoft Windows XP z zainstalowanym pakietem Microsoft .NET Framework version 2.0. Wyniki działania aplikacji *Rough Classifier* przedstawiono w tabeli 11. Jako, że zbiory *data1* i *data2* mają różną liczbę obiektów (użytkowników), więc w celu umożliwienia porównania wartości funkcji odległości  $\delta$  jej wartości należy znormalizować przez podzielenie jej wartości zdefiniowanej w punkcie 7.2 przez kwadrat liczby użytkowników podzielony przez 2.

Rezultaty przeprowadzonych badań pokazały, że we wszystkich przypadkach przybliżony algorytm RC-3 pozwolił na uzyskanie wyników dokładnie takich samych jak algorytm *brute-force* przeszukujący całą przestrzeń rozwiązań z jednoczesnym, około 40-krotnym skróceniem czasu działania. Czas działania algorytmu RC-3 jest dostatecznie krótki, aby można go było efektywnie wykorzystywać jako działający w tle całego systemu rekomendacyjnego.

Tabela 11. Wyniki działania aplikacji *Rough Classifier* [Nguyen 2006]

| Zbiór danych | Identyfikatory atrybutów,<br>które nie były analizowane | Identyfikatory atrybutów<br>generujących podział $Q$ | Liczba wartości dla atrybutów<br>o ciągłej dziedzinie | Identyfikatory atrybutów<br>generujące wynikowy podział $P_D$ | Znormalizowana odległość $\delta(P_D, Q)$ | Czas działania algorytmu<br>RC-3 (w s) | Identyfikatory atrybutów<br>generujące wynikowy podział $P_{D'}$<br>metodą brute-force | Znormalizowana odległość $\delta(P_{D'}, Q)$ | Czas działania metody<br><i>brute-force</i> (w s) |
|--------------|---------------------------------------------------------|------------------------------------------------------|-------------------------------------------------------|---------------------------------------------------------------|-------------------------------------------|----------------------------------------|----------------------------------------------------------------------------------------|----------------------------------------------|---------------------------------------------------|
| data2        | 1, 10, 11                                               | 0, 14                                                | 3                                                     | 5, 12                                                         | 0,44049                                   | 90,3125                                | 5, 12                                                                                  | 0,44049                                      | 3554,8437                                         |
| data1        | 1, 10, 11                                               | 0, 14                                                | 3                                                     | 5, 12                                                         | 0,45897                                   | 351,8125                               | 5, 12                                                                                  | 0,44680                                      | 14750,3125                                        |
| data2        | 4, 7, 12                                                | 9, 10, 11                                            | 5                                                     | 5                                                             | 0,41814                                   | 54,109375                              | 5                                                                                      | 0,41814                                      | 1872,4375                                         |
| data1        | 4, 7, 12                                                | 9, 10, 11                                            | 5                                                     | 5                                                             | 0,42061                                   | 220,296875                             | 5                                                                                      | 0,42061                                      | 7785,6875                                         |
| data2        | 2                                                       | 1, 3, 14                                             | 3                                                     | 0, 4, 5, 6, 7,<br>8, 9, 10,<br>11, 12, 13                     | 0,08248                                   | 213,2968                               | 0, 4, 5, 6, 7,<br>8, 9, 10,<br>11, 12, 13                                              | 0,08248                                      | 46971,0312                                        |

## 7.6. Podsumowanie

Przedstawiono wykorzystanie klasyfikacji przybliżonej do wyznaczenia optymalnego profilu użytkownika. Rozwiązanie tego problemu jest szczególnie istotne w udoskonalaniu działających systemów rekomendacyjnych. Nawet najlepszy projektant nie jest w stanie przewidzieć tego czy zestaw danych zbieranych o użytkowniku jest najbardziej korzystny ze względu na dostarczane przez system rekomendacje.

Dane zbierane o użytkowniku, czyli zbiór atrybutów może być zbyt obszerny i wtedy mamy do czynienia z redundantnymi danymi, które ze względu na dostarczane rekomendacje są zupełnie zbędne. W przeciwnym razie dane zbierane o użytkownikach mogą być zbyt ubogie, a co za tym idzie niedające żadnych możliwości pozwalających na dostarczenie rekomendacji na zadowalającym poziomie.

W ramach prezentacji metody wykorzystania klasyfikacji przybliżonej przedstawiono koncepcję oraz podstawowe pojęcia klasyfikacji przybliżonej na podstawie prac Pawlaka [Pawlak 1983] i [Pawlak 1999]. Przedstawiono również poprawioną wersję wyników badań w pracach [Nguyen 2005] i [Nguyen 2006]: sformułowanie trzech problemów klasyfikacji przybliżonej (RC-1–RC-3), dwa twierdzenia dotyczące własności odległości między podziałami służące do sformułowania rozwiązań przedstawionych problemów oraz algorytmy przybliżone rozwiązania tych problemów. Problemy RC-1 i RC-2 ze względu na ograniczenia dotyczące podziału wynikowego, w praktycznych zastosowaniach mogą być trudne do osiągnięcia, stąd weryfikację eksperymentalną wykonano jedynie dla rozwiązań problemu RC-3. Najpierw przedstawiono rozwiązanie RC-3 dla rzeczywistych danych uzyskanych w eksperymencie z udziałem rzeczywistych użytkowników systemu *Megane*, przedstawionego bliżej w rozdziale 5. Rozwiązanie to uzyskano dwoma metodami, za pomocą zaproponowanego algorytmu przybliżonego oraz za pomocą binarnego algorytmu genetycznego. Zaproponowany algorytm przybliżony o złożoności obliczeniowej  $O(m^2n^2)$ , gdzie  $m = 5$  jest liczbą atrybutów, a  $n = 8$  liczbą użytkowników, nie daje w wyniku podziału o minimalnej odległości od zadanego podziału, natomiast binarny algorytm genetyczny o populacji i liczbie generacji 10 znajduje podziały o minimalnej odległości od zadanego podziału.

Dane systemu *Megane* dotyczą bardzo niewielkiej liczby użytkowników, stąd weryfikację przedstawionej metody rozwiązania problemu RC-3 przedstawiono również dla dwóch zbiorów danych *Adult* [Adult 1998], zawierających rzeczywiste dane demograficzne dla 30 162 i 15 059 osób i 15 atrybutów. W eksperymencie część tych atrybutów była wykorzystywana do wygenerowania docelowego podziału  $Q$ . W celu weryfikacji przybliżonego algorytmu zaimplementowano również algorytm przeszukiwania całej przestrzeni rozwiązań. We wszystkich analizowanych przypadkach algorytm przybliżony dawał rozwiązanie takie samo jak algorytm przeszukiwania całej przestrzeni rozwiązań jednak w czasie 40-krotnie krótszym.

Indywidualnym wkładem Autora w przedstawionych w niniejszym rozdziale wynikach prac było m.in. zaproponowanie dziedziny zastosowań klasyfikacji przybliżonej, przedstawienie własnych dowodów twierdzeń, określenie własności zaproponowanych

algorytmów rozwiązania problemów RC-1–RC-3, zaproponowanie zastosowania algorytmu genetycznego do rozwiązania RC-3, opracowanie algorytmu przeszukiwania całej przestrzeni rozwiązań pomijającego przeszukiwanie gałęzi, które nie prowadzą do rozwiązania oraz przeprowadzenie oraz opracowanie wyników eksperymentów.



## 8. Uwagi końcowe

W ostatnim czasie odnotowano znaczny postęp zarówno dotyczący liczby zastosowań, jak i stosowanych metod oraz jakości dostarczanych rekomendacji. Dalszy rozwój systemów rekomendacyjnych polega na zastosowaniu różnego rodzaju rozszerzeń, takich jak [Adomavicius 2005]: pogłębione modelowanie użytkowników i rekomendowanych elementów, uwzględnienie informacji kontekstowej w rekomendacji, wsparcie dla wielokryterialnych ocen oraz zapewnienie bardziej fleksybilnych i mniej intruzywnych procesów rekomendacyjnych.

Wskazywane w literaturze potencjalne kierunki rozwoju systemów rekomendacyjnych zostały w większości wykorzystane w niniejszej monografii. W implementacji systemu z rekomendacją interfejsu wykorzystano metody consensusu opisanego w rozdziale 5 zastosowano wielokryterialną ocenę satysfakcji użytkowanego interfejsu użytkownika. Przedstawiona w pracy koncepcja dostarczenia spersonalizowanych interfejsów nie jest zbyt intruzyjna. Wymaga od użytkownika podania jedynie kilku danych demograficznych, które służą do znalezienia podobnych użytkowników. Większość z użytkowników nie musi żmudnie personalizować manualnie swój interfejs lub oceniać poszczególne jego komponenty, aby uzyskać zaadaptowany na jego potrzeby interfejs użytkownika.

Potrzebę pogłębionego modelowania użytkownika wykazały badania nad systemem *Reporter*, który został przedstawiony w punkcie 6.1. System *Reporter* wykorzystuje ideę sztucznego systemu immunologicznego (AIS) do rekomendacji wiadomości oraz interfejsu użytkownika. Badania nad tym systemem pokazały, że podobieństwo użytkowników oparte na ocenach dotyczących preferencji wiadomości pozwala na efektywne rekomendowanie wiadomości, natomiast rekomendacje interfejsów użytkownika oparte na tej samej podstawie nie przynoszą podobnych efektów. Przedstawiono zastosowanie metafory kolonii mrówek do rekomendacji interfejsów użytkownika, która wypełnia dwa z zaprezentowanych postulatów, gdyż prezentuje bardziej pogłębione modelowanie zarówno użytkownika jak i rekomendowanego elementu, czyli interfejsu użytkownika, w postaci struktury drzewiastej umożliwiającej w bardzo naturalny sposób przedstawienie zależności, które występują między poszczególnymi elementami interfejsu. Metafora kolonii mrówek umożliwia realizację procesu rekomendacji w fleksybilny sposób, gdyż w zależności od istniejących danych pozwala na równoczesne zastosowanie trzech podstawowych metod filtrowania: DF, CF i CBF. Metodę tę do rekomendacji zweryfikowano na przykładzie rekomendacji filmów. Zaprezentowano metaforę kolonii mrówek

podobnie jak zastosowanie metod consensusu stanowi wkład Autora do rozwoju dziedziny systemów rekomendacyjnych.

Przedstawiono wykorzystanie klasyfikacji przybliżonej do wyznaczenia optymalnego profilu użytkownika. Rozwiązanie problemu wyznaczenia optymalnego zbioru atrybutów opisujących użytkowników systemów rekomendacyjnych jest bardzo istotne w budowie efektywnych systemów rekomendacyjnych. Stanowi ono również wypełnienie ostatniego z postulatów dotyczących tworzenia nowoczesnych systemów rekomendacyjnych [Adomavicius 2005], a mianowicie może się przyczynić do zmniejszenia poziomu intruzyjności procesów rekomendacyjnych, co wiąże się ze zmniejszeniem liczby informacji, które musi dostarczyć użytkownik, aby otrzymać rekomendację na zadowalającym go poziomie efektywności. Nawet w przypadku gdy określone dane są zbierane automatycznie, wówczas gdy nie są one niezbędne w procesie rekomendacji i nie muszą one być zapisywane, co należy traktować jako zabieg pozytywny do zachowania prywatności przez użytkowników.

Na uwagę zasługuje także to, iż przedstawiony przegląd metod zapewnienia użyteczności systemów webowych został uzupełniony o własne doświadczenia związane z badaniem użytkowników, w tym z wykorzystaniem nowoczesnych metod bazujących na śledzeniu wzroku. Przedstawiono również sposoby zautomatyzowania badania użyteczności systemów webowych.

Dzięki wykorzystaniu różnego typu systemów webowych z roku na rok rośnie rynek e-Commerce. Jednym z podstawowych celów każdego z dysponentów tego typu systemów jest zdobycie i utrzymanie określonej części użytkowników docelowych. Na globalizującym się rynku ich zawartość oraz oferowane usługi są porównywalne, stąd konieczność szukania przewagi, np. na polu użyteczności, w tym satysfakcji użytkownika z korzystania z danego interfejsu. Prezentowane w niniejszej pracy metody rekomendacji interfejsów mogą dostarczyć wygodnych narzędzi do tworzenia bardziej użytecznych interfejsów użytkownika.

## Literatura

- [Abdi 2007] Abdi H., *Kendall rank correlation*. Encyclopedia of Measurement and Statistics. Thousand Oaks (CA): Sage 2007. <http://www.utdallas.edu/~herve/Abdi-KendallCorrelation2007-pretty.pdf>.
- [Adamczyk 2003] Adamczyk J., *Zbawienna personalizacja – czyli jak zdobyć lojalność e-klienta*. Internet 08.01.2003. Pobrano lipiec 2007 <http://www.e-marketing.pl/artyk/artyk53.php>.
- [Adomavicius 2005] Adomavicius G, Tuzhilin A., *Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*. IEEE Transactions on Knowledge and Data Engineering, 17(6), 2005, 734–749.
- [Adult 1998] Adult Data Set. Pobrano grudzień 2005 <http://rss.acs.unt.edu/Rdoc/library/arules/html/Adult.html>.
- [Age Company 2007] Age Company Ltd., Gaming apes wow zoogoers. April 12, 2007. Pobrano wrzesień 2007 <http://www.theage.com.au/news/technology/gaming-apes-wow-zoogoers/2007/04/12/1175971232506.html#>.
- [Agrawal 1993] Agrawal R., Imielinski T., Swami A.N., *Mining Association Rules between Sets of Items in Large Database*, SIGMOD, 22(2), 1993, 207–216.
- [Agrawal 1994] Agrawal R., Srikant R., *Fast Algorithms for Mining Association Rules*, Proc. of VLDB, Chile, Sep. 12–15 1994, 487–99.
- [Aizerman 1985] Aizerman M.A., *New Problems in the General Choice Theory*. Social Choice Welfare 2, 1985, 235–382.
- [Aktas 2004] Aktas M.S., Pierce M., Fox G.C., Leake D., *A Web based Conversational Case-Based Recommender System for Ontology aided Metadata Discovery*. Proc. of the Fifth IEEE/ACM international Workshop on Grid Computing (November 08–08, 2004). International Conference on Grid Computing. IEEE Computer Society, Washington, DC, 2004, 69–75. <http://dx.doi.org/10.1109/GRID.2004.6>
- [Allaire 2002] Allaire J., *Macromedia Flash MX – A next-generation rich client*. Macromedia Inc. March 2002. Pobrano grudzień 2008 <http://www.adobe.com/devnet/flash/whitepapers/rich-client.pdf>.
- [ASL 2005] ASL (Applied Science Laboratories), *Eye Tracking System Instructions, ASL Eye-Trac 6000 Head Mounted Optics*. EyeTracHMOManual.pdf Version 1.01 04/05/2005. Applied Science Group, Inc. 2005.
- [ASL 2006] ASL (Applied Science Laboratories), *Eye Tracking System Instructions ASL Eye-Trac 6000 Pan/Tilt Optics*. EyeTracPanTiltManual.pdf, ASL Version 1.04 01/17/2006. Applied Science Group, Inc., 2006.

- [Aslin 2004] Aslin R.N., McMurray B., *Automated corneal-reflection eye-tracking in infancy: Methodological developments and applications to cognition*. INFANCY, 6(2), 2004. 155–163 [http://www.bcs.rochester.edu/people/aslin/pdfs/Aslin\\_McMurray2004.pdf](http://www.bcs.rochester.edu/people/aslin/pdfs/Aslin_McMurray2004.pdf).
- [ASP 2007] ASP.NET, *Introduction*. Pobrano lipiec 2007 [http://www.w3schools.com/aspnet/aspnet\\_intro.asp](http://www.w3schools.com/aspnet/aspnet_intro.asp).
- [Atkinson 1989] Atkinson M., Bancilhon F., DeWitt D., Dittrich K., Maier D., Zdonik S., *The Object-Oriented Database System Manifesto*. [In:] Proceedings of the First International Conference on Deductive and Object-Oriented Databases, Kyoto, Japan. 1989, 223–240.
- [Bar-Cohen 2003] Bar-Cohen B., Breazeal C.L., *Biologically Inspired Intelligent Robots*. SPIE Press Monograph Vol. PM122, 2003.
- [Basili 2003] Basili R., Pazienza M.T., Zanzotto F.M., *Exploiting the feature vector model for learning linguistic representations of relational concepts*. [In:] Proceedings of the International Workshop & Tutorial on Adaptive Text Extraction and Mining held in conjunction with the 14th European Conference on Machine Learning and the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases 22 September 2003 Cavtat–Dubrovnik (Croatia), 2–10.
- [Bednarek 2002] Bednarek J., *Media w Nauczaniu*. Mikom: Warszawa 2002.
- [Berners-Lee 1996] Berners-Lee T., Fielding R., Frystyk H., *Hypertext Transfer Protocol – HTTP/1.0*, Network Working Group RFC 1945, May, 1996. Pobrano grudzień 2008 <http://www.w3.org/Protocols/rfc1945/rfc1945>.
- [Bichindaritz 2006] Bichindaritz I., Marling C., *Case-based reasoning in the health sciences: What's next?*, J. Artificial Intelligence in Medicine, 36(2), 2006, 127–135.
- [Blum 2005] Blum C., *Ant colony optimization: Introduction and recent trends*. Physics of Life Reviews 2(4), 2005, 353–373.
- [Breese 1998] Breese J., Heckerman D., Kadie C., *Empirical analysis of predictive algorithms for collaborative filtering*. Technical report, Microsoft Research, October 1998. <ftp://ftp.research.microsoft.com/pub/tr/tr-98-12.pdf>.
- [Brown 1998] Brown S.M., Santos E., Banks S.B., Oxley M.E., *Using explicit requirements and metrics for interface agent user model for interface agent user model correction*. [In:] Proc. of the second international conference on Autonomous Agents, Minneapolis, Minnesota, USA, 1998, 1–7.
- [Boudjeloud 2005] Boudjeloud L., Poulet F., *Attribute Selection for High Dimensional Data Clustering*. [In:] Proc. of the Conference on Applied Stochastic Models and Data Analysis (ASM-DA 2005), 2005, 387–394. <http://asmda2005.enst-bretagne.fr/IMG/pdf/proceedings/387.pdf>.
- [Bush 1945] Bush V., *As We May Think*. The Atlantic. July 1945. <http://www.theatlantic.com/doc/194507/bush>.
- [CGI 95] Common Gateway Interface. Pobrano lipiec 2007 <http://hoohoo.ncsa.uiuc.edu/cgi/intro.html>
- [Chapman 2004] Chapman N., Chapman J., *Digital Media*. John Wiley & Sons: Chichester 2004.
- [Chen 1998] Chen L., Sycara K.P., *WebMate: A Personal Agent for Browsing and Searching*. [In:] Proc. of the second international conference on Autonomous agents, Minneapolis, Minnesota, United States 1998, 132–139.
- [Chen 2000] Chen Z., Meng X., Zhu B., Fowler R., *WebSail: From On-Line Learning to Web Search*. [In:] Proc. of the 2000 International Conference on Web Information Systems Engineering, 1, 2000, 206–213. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.30.831&rep=rep1&type=pdf>.

- [Chen 2004] Chen Q., Aickelin U., *Movie Recommendation Systems Using An Artificial Immune System*. [In:] Poster Proc. of ACDM 2004 Engineers' House, Bristol, UK. <http://arxiv.org/ftp/arxiv/papers/0801/0801.4287.pdf>.
- [Choroś 1992] Choroś K. (red.), Karpowicz G., Siemiński A., Sonecki J., Trawiński B., *Mikrokomputerowe systemy biblioteczno-informacyjne*. Wydawnictwo Politechniki Wrocławskiej, Wrocław 1992.
- [Christmann 2006] Christmann O., Carbonell N., *Browsing through 3D representations of unstructured picture collections: an empirical study*. [In:] Proc. of ACM Working Conference on Advanced Visual Interfaces, Venezia, Italy, May 23–26, 2006, 369–372.
- [Cichosz 2000] Cichosz P., *Systemy uczące się*. WNT: Warszawa 2000.
- [Cliff 2003] Cliff D., *Biologically-Inspired Computing Approaches to Cognitive Systems: a Partial Tour of the Literature*. HP Laboratories Bristol. HPL-2003-11, April 14, 2003. <http://www.hpl.hp.com/techreports/2003/HPL-2003-11.pdf>
- [Cohen 2003] Cohen W.W., Ravikuma P., Fienberg S.E., *A Comparison of String Distance Metrics for Name-Matching Tasks*. [In:] Proc. of IIWEB, 2003, 73–78. <http://www.cs.cmu.edu/~wcohen/postscript/ijcai-ws-2003.pdf>
- [Collins 1969] Collins A.M., Quillian M.R., *Retrieval time from semantic memory*. Journal of Verbal Learning and Verbal Behaviour, 8 (2), 1969, 240–247.
- [Colomi 1996] Colomi A., Dorigo M., Maoli F., Maniezzo V., Righini G., Trubian M., *Heuristics from Nature for Hard Combinatorial Optimization Problems*. International Transactions in Operational Research, 3(1), 1996, 1–21. <http://citeseer.ist.psu.edu/cache/papers/cs/20003/http://zSzzSzdsp.jpl.nasa.gov/zSzmemberszSzpaymanzSzswarmzSzcolorni96-itor.pdf/colorni96heuristics.pdf>
- [Daniłowicz 1992] Daniłowicz Cz., *Modele systemów informacji uwzględniające preferencje użytkowników końcowych*. Prace Naukowe BGiOINT Politechniki Wrocławskiej. Seria: Monografie 6 (3). Wydawnictwo Politechniki Wrocławskiej, Wrocław 1992.
- [Daniłowicz 2002] Daniłowicz Cz., Nguyen N.T., Jankowski Ł., *Metody wyboru reprezentacji stanu wiedzy agentów w systemach multiagenckich*. Oficyna Wydawnicza Politechniki Wrocławskiej: Wrocław 2002.
- [Date 1981] Date C.J., *Wprowadzenie do baz danych*. WNT, Warszawa 1981.
- [Dattola 1968] Dattola R.T., *A fast algorithm for automatic classification*. Report ISR–14 to the National Science Foundation, Section V, Cornell University, Department of Computer Science 1968.
- [Day 1988] Day W.H.E., *Consensus methods as tools for data analysis*. [In:] Bock H.H. (eds.), *Classification and related Methods of Data Analysis*. Proceedings of IFCS'87, North-Holland, 1988, 317–324.
- [Dąbrowski 1978] Dąbrowski M., Laus-Maczyńska K., *Metody wyszukiwania i klasyfikacji informacji*. WNT, Warszawa 1978.
- [Deneubourg 1990] Deneubourg J.L., Aron S., Goss S., Pasteels J.M., *The self-organizing exploratory pattern of the argentine ant*. Journal of Insect Behavior, 3, 1990, 159–168. <http://www.ulb.ac.be/sciences/use/publications/JLD/76.pdf>.
- [O'Donovan 2003] O'Donovan J., Dunnion J., *A Comparison of Collaborative Recommendation Algorithms over Diverse Data Types*. [In:] Proc. of 14th Irish Conference on Artificial Intelligence and Cognitive Science (AICS), Dublin 2003, 101–203.

- [O'Donovan 2004] O'Donovan J., Dunnion J., *Adaptive Recommendation: Putting The Best Foot Forward*. [In:] Proc. of 3rd International Symposium on Information and Communication Technologies (ISICT), Las Vegas, Nevada, USA, June 16–18 2004, 109–114.
- [Dorigo 1999] Dorigo M., Di Caro G., *Ant Algorithms for Discrete Optimization*. Artificial Life. 5(2), 1999, 137–172. [http://www.idsia.ch/~luca/ij\\_23-alife99.pdf](http://www.idsia.ch/~luca/ij_23-alife99.pdf)
- [Elliott 2008] Elliott D., Rutherford J., Erickson J.S., *A Recommender System for the DSpace Open Repository Platform*. Web Services and Systems Laboratory HP Laboratories, Bristol, HPL–2008–21, April 8, 2008.
- [Engelbart 1962] Engelbart D.C., *Augmenting Human Intellect: A Conceptual Framework*, AFOSR-3233 Summary Report, SRI Project No. 3579. <http://www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html>.
- [Feizabadi 1998] Feizabadi S., *History of the World Wide Web*. [In:] Abrams M. (ed.), *World Wide Web – Beyond the Basics*, Prentice Hall: 1998. <http://ei.cs.vt.edu/~wwwbtb/book/chap1/index.html>.
- [Fellbaum 2001] Fellbaum C., Palmer M., Dang H.T., Delfs L., Wolf S., *Manual and automatic semantic annotation with WordNet*. [In:] Proc. of the NAACL 2001 Workshop on WordNet and Other Lexical Resources, Pittsburgh, June 2001, 1–7.
- [Frøkjær 2000] Frøkjær E., Hertzum M., Hornbæk K., *Measuring Usability: Are Effectiveness, Efficiency, and Satisfaction Really Correlated?* [In:] Proc. of the ACM CHI 2000 Conference on Human Factors in Computing Systems, The Hague, The Netherlands, April 1–6, 2000, 345–352.
- [Garret 2005] Garrett J.J., *Ajax: A New Approach to Web Applications*. 2005. Pobrano lipiec 2007 z adresu: <http://www.javalooby.org/articles/ajax/>
- [Garret 2005a] Garrett S.M., *How Do We Evaluate Artificial Immune Systems?*. Evolutionary Computation 13(2), 2005, 145–178. <http://www.mitpressjournals.org/doi/pdf/10.1162/1063656054088512?cookieSet=1>.
- [Gawrysiak 1998] Gawrysiak P., Mańkowski P., Uchański A., *Biblia komputerowego gracza*. Iskry, Warszawa 1998.
- [Good 1999] Good N., Schafer J.B., Konstan J., Borchers A., Sarwar B., Herlocker J., Riedl J., *Combining Collaborative Filtering with Personal Agents for Better Recommendations*. [In:] Proc. of the 1999 Conference of the American Association of Artificial Intelligence (AAAI-99), 439–446.
- [Google 2004] *Google Technology*, 2004. Pobrano lipiec 2006 <http://www.google.pl/intl/pl/technology/index.html>.
- [Google 2007] *Google Web Toolkit – Build AJAX apps in the Java language*. Pobrano lipiec 2007 z adresu: <http://code.google.com/webtoolkit/>.
- [Grupe 1998] Grupe F.H., Urwiler R., Ramarapu N.K., Owrang M., *Case-based reasoning; CBR; Software development; Function point analysis; Systems analysis; Systems design; Software reusability*. Information and Software Technology 40(9), 1998, 493–499.
- [Grouplens 2008] Grouplens Research, *MovieLens Data Sets*. Pobrano czerwiec 2008 <http://www.grouplens.org/node/73>.
- [Hackos 1998] Hackos J.T., Redish J.D., *User and Task Analysis for Interface Design*. Appendix C. John Wiley and Sons, New York, 1998, 457–478.
- [Hardman 2008] Hardman L., Pemberton S., *The Path to Web n+1*. ERCIM News, (72), 2008, 16–17.
- [Harmelen 2008] Harmelen F., *Which Future Web?*, ERCIM News, (72), 2008, 18–19.

- [Haupt 2004] Haupt R.L., Haupt E.H., *Practical Genetic Algorithms*. Wiley-Interscience Hoboken, New Jersey 2004.
- [Herlocker 1999] Herlocker J.L., Konstan J.A., Borchers A., Riedl J., *An Algorithmic Framework for Performing Collaborative Filtering*, [In:] Proc. of the 22<sup>nd</sup> Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, August Berkeley, CA, 1999, USA, 15–19. <http://www.grouplens.org/papers/pdf/algs.pdf>.
- [Herlocker 2004] Herlocker J.L., Konstan J.A., Terveen L.G., Riedl J., *Evaluating Collaborative Filtering Recommender Systems*. ACM Transactions on Information Systems, 22, 2004, 5–53.
- [Hewett 1992] Hewett T., Baecker R., Card S., Carey T., Gasen J., Mantei M., Perlman G., Strong G., Verplank W., *ACM SIGCHI Curricula for Human-Computer Interaction*. Report of the ACM SIGCHI Curriculum Development Group, ACM: 1992. <http://acm.org/sigchi/cdg/>
- [Hochbaum 1982] Hochbaum D., Steele J.M., *Steinhaus's Geometric Location Problem for Random Samples in the Plane*. Advanced Applied Probability, 14 (1982), 56–67.
- [Holland 1975] Holland J.H., *Adaptation in natural and artificial systems*. Ann Arbor, The University of Michigan Press, 1975.
- [Holt 2002] Holt J., *UML (Unified Modelling Language) for Systems Engineers*. Lavoisier 2002.
- [Hom 1998] Hom J., *The Usability Methods Toolbox*. 1998. Pobrano lipiec 2008 <http://jthom.best.vwh.net/usability/>.
- [ISO 1996] International Standard ISO 9241-10. *Ergonomic requirements for office work with visual display terminals (VDTs) – Part 10: Dialogue principles*. ISO 1996.
- [ISO 1997] International Standard ISO 9241-11. *Ergonomic requirements for office work with visual display terminals (VDT's) – Part 11: Guidance on Usability*. ISO 1997.
- [Iyer 2005] Iyer L., *Computer software games for cats*. Pura Magazine 03, 2005. [http://www.pets-pourri.com/latestissue03\\_05.htm](http://www.pets-pourri.com/latestissue03_05.htm).
- [Jakubowski 1996] Jakubowski Ł., Katarzyniak R., Sobecki J., *Implementacja katalogu on-line z wykorzystaniem WAIS*. [W:] *Zintegrowane systemy biblioteczne w aspekcie stworzenia jednolitego systemu bibliotecznego*. Materiały konferencyjne. Biblioteka Główna PBiałost. Białystok–Supraśl, 17–19 czerwca 1996. Białystok, Wydaw. PBiałost. 1996, 137–148.
- [Jakubowski 1997] Jakubowski Ł., Sobecki J., *Problemy implementacji pełnotekstowych baz raportów technicznych w sieci Internet*. [W:] *Infobazy '97*. Bazy danych dla nauki. Centrum Informatyczne TASK, Instytut Oceanologii PAN, PGdań. Gdańsk, 23–25 czerwca 1997. Gdańsk, Centr. Inform. TASK PGdań. 1997, 380–390.
- [Kanago 2002] Kanungo T., Mount D.M., Netanyahu N.S., Piatko C., Silverman R., Wu A.Y., *An Efficient k-means clustering algorithm: analysis and implementation*. IEEE Transactions On Pattern Analysis And Machine Intelligence, 24(7), 2002, 881–892.
- [Karpiak 1996] Karpiak-Kozińska D., Katarzyniak R., Kazienko P., Leśniak D., Sobecki J., Trawiński B., Daniłowicz Cz., *Sieciowe i multimedialne systemy informacyjne*. Praca zbiorowa pod red. Czesława Daniłowicza. Wrocław, Zakład Systemów Informacyjnych Wydziału Informatyki i Zarządzania PWroc., 1996.
- [Katarzyniak 1997] Katarzyniak R., Kazienko P., Sobecki J., *Wybrane problemy wyszukiwania informacji w sieci Internet*. [W:] *IV Krajowe Forum Informacji Naukowej i Technicznej*. Materiały konferencyjne. Polskie Towarzystwo Informacji Naukowej. Zakopane, 2–5 września 1997, 286–295.
- [Katz-Haas 1998] Katz-Haas R., *Ten Guidelines for User-Centered Web Design*. Usability Interface, 5(1), 1998. [http://www.stcsig.org/usability/topics/articles/ucd%20\\_web\\_devel.html#how\\_to](http://www.stcsig.org/usability/topics/articles/ucd%20_web_devel.html#how_to).



- [Kazienko 2004] Kazienko P., Sobecki J., *XML-Based Learning Scenario Representation and Presentation in the Adaptive e-Learning Environment*. Lecture Notes in Computer Science, 3029, 2004, 967–976.
- [Knuckles 2005] Knuckles C.D., *Web Applications. Concepts & Real World Design*. John Wiley & Sons, New York 2005.
- [Kobsa 2001] Kobsa A., Koenemann J., Pohl W., *Personalized Hypermedia Presentation Techniques for Improving Online Customer Relationships*. The Knowledge Engineering Review 16(2), 2001, 111–155.
- [Kobsa 2002] Kobsa A., *Personalized Hypermedia and International Privacy*. Communications of the ACM 45(5), 2002, 64–67.
- [Kołcio 1995] Kołcio D., Sobecki J., *Wykorzystanie WWW w bibliotekach i ośrodkach informacji naukowo-technicznej*. Prakt. Teor. Inf. Nauk. Tech. 3(4), 1995, 8–13.
- [Kotonya 1998] Kotonya G., Sommerville I., *Requirements Engineering Processes and Techniques*. Wiley 1998.
- [Kowalczyk 2001] Kowalczyk M., *Moje prywatnie Idaho*. Internet Standard 1, 2001. <http://www.idg.pl/artykuly/7721.html>.
- [Kristol 1997] Kristol D., Montulli L., *HTTP State Management Mechanism*, Network Working Group RFC 2109, February 1997. Pobrano grudzień 2008 <http://www.ietf.org/rfc/rfc2109.txt>.
- [Krug 2000] Krug S., *Don't Make Me Think*, New Riders Press, 2000, 5.
- [Kukla 2001] Kukla E., Nguyen N.T., Sobecki J., *Towards the model of tutoring strategy selection based on consensus procedure in hypermedia*. [In:] Proc. of the 14th International Conference on Systems Science. Eds Z. Bubnicki, A. Grzech. Wrocław, 11–14 September 2001. Vol. 2. *Systems and control engineering. Operation and manufacturing systems. Knowledge engineering and intelligent systems*. Wrocław, Oficyna Wydaw. PWr. 2001, 404–411.
- [Kukla 2002] Kukla E., Nguyen N.T., Sobecki J., *The consensus based tutoring strategy selection in CAL systems*. World Transactions on Engineering Education 1(1), 2002, 95–98.
- [Kukla 2003] Kukla E., Nguyen N.T., Sobecki J., Danilowicz C., Lenar M., *A Model Conception for Learner Profile Construction and Determination of Optimal Scenario in Intelligent Learning System*. Lecture Notes in Computer Science 2774, 2003, 1216–1222.
- [Kukla 2004] Kukla E., Nguyen N.T., Sobecki J., Danilowicz C., Lenar M., *A model conception for learner profile construction and determination of optimal scenario in intelligent learning systems*. ITSE – International Journal of Interactive Technology and Smart Education 1(3), 2004, 171–184.
- [Kukla 2004a] Kukla E., Nguyen N.T., Sobecki J., Danilowicz C., Lenar M., *Determination of Learning Scenarios in Intelligent Web-Based Learning Environment*. Lecture Notes in Computer Science 3009, 2004, 759–768.
- [Kuropka 2006] Kuropka I., Pisz Z., *Koncepcja budowy miernika syntetycznego*. Akademia Ekonomiczna we Wrocławiu, 2006. Pobrano lipiec 2007 <http://www.genderindex.pl/downloads/KoncepcjaWskaznikaGI.doc>
- [Lenar 2007] Lenar M., Sobecki J., *Using Recommendation to Improve Negotiations in Agent-based Systems*. Journal of Universal Computer Science 13 (2), 2007, 267–285.
- [Lennon 1997] Lennon J.A., *Hypermedia Systems and Applications. World Wide Web and Beyond*. Springer, Berlin 1997.
- [Lieberman 1997] Lieberman H., *Autonomous Interface Agents*. [In:] Proc. CHI 97, ACM, 1997, 67–74.

- [Lloyd 1982] Lloyd S.P., *Least squares quantization in PCM*. IEEE Trans. Inform. Theory, 28, 1982, 129–137. first published S. Lloyd, *Last square quantization in PCM's*. Bell Telephone Laboratories Paper (1957).
- [Loosley 2006] Loosley C., *Rich Internet Applications: Design, Measurement, and Management Challenges*, Keynote Systems, 2006. Pobrano grudzień 2008 [http://www.keynote.com/docs/whitepapers/RichInternet\\_5.pdf](http://www.keynote.com/docs/whitepapers/RichInternet_5.pdf)
- [Maglio 2000] Maglio P.P., Barrett R., Campbell C.S., Selker T., *SUITOR: an attentive information system*. [In:] Proc. of the 2000 Int Conf on Intelligent User Interfaces, 2000, 169–176.
- [Magnini 2002] Magnini B., Strapparava C., *User modelling for news web sites with content-based techniques*. [In:] Proceedings WWW–2002, the Eleventh International World Wide Web Conference, Poster session, Honolulu, Hawaii, USA. 2002.
- [Maloof 2000] Maloof M., Michalski R., *Selecting examples for partial memory learning*. Machine Learning 41, 2000, 27–52.
- [Mamdani 1974] Mamdani E.H., *Application of fuzzy algorithms for the control of a simple dynamic plant*. [In:] Proc IEEE 1974, 121–159.
- [Manning 2008] Manning C.D., Raghavan P., Schuetze H., *Introduction to Information Retrieval*, Cambridge University Press, 2008.
- [Massa 2004] Massa P., Avesani P., *Trust-aware Collaborative Filtering for Recommender Systems*, [In:] Proc. of ITC-iRST, Povo, Italy, 2004. [http://sra.itc.it/people/massa/publications/massa\\_paolo\\_coopis\\_2004\\_trust-aware\\_Collaborative\\_Filtering\\_for\\_Recommender\\_Systems.pdf](http://sra.itc.it/people/massa/publications/massa_paolo_coopis_2004_trust-aware_Collaborative_Filtering_for_Recommender_Systems.pdf)
- [Maybury 1997] Maybury M.T. (ed.), *Intelligent Multimedia Information Retrieval*. The MIT Press, Cambridge 1997.
- [McLaughlin 2004] McLaughlin M.M., Herlocker J.L., *A Collaborative Filtering Algorithm and Evaluation Metric that Accurately Model the User Experience*. [In:] Proc. of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. Sheffield, United Kingdom 2004, 329–336.
- [Metz 2007] Metz C., *Web 2.0*. PC Magazine 14.03.2007. Pobrano lipiec 2007 <http://www.pcmag.com/article2/0,1895,2102852,00.asp>.
- [Microsoft 1999] Microsoft Corporation, *Windows User Experience: Official Guidelines for User Interface Developers and Designers*. Microsoft Professional Editions. Microsoft Press, US 1999.
- [Microsoft 2007] Microsoft Corporation, *Windows Vista User Experience Guidelines*. Microsoft Corporation 2007. Pobrano lipiec 2008 <http://download.microsoft.com/download/e/1/9/e191fd8c-bce8-4dba-a9d5-2d4e3f3ec1d3/ux%20guide.pdf>.
- [Miligram 94] Milgram P., Kishino F., *Taxonomy of Mixed Reality Visual Displays*. EICE Transactions on Information Systems, E77-D(12), 1994. [http://gypsy.rose.utoronto.ca/people/paul\\_dir/IEICE94/ieice.html](http://gypsy.rose.utoronto.ca/people/paul_dir/IEICE94/ieice.html).
- [Miller 1997] Miller B., Riedl J., Konstan J., *Experiences with GroupLens: Making Usenet Useful Again*, Usenix Winter Technical Conference, 1, 1997, 1–17. Pobrano grudzień 2008 <http://www.grouplens.org/papers/pdf/usenix97.pdf>.
- [Miniwatts 2008] Miniwatts Marketing Groups, *World Internet Usage Statistics News and World Population Stats*. Pobrano lipiec 2008 <http://www.internetworldstats.com/stats.htm>.
- [Miyahara 2000] Miyahara K., Pazzani M.J., *Collaborative Filtering with the Simple Bayesian Classifier*. [In:] Proc. of Pacific Rim International Conference on Artificial Intelligence, 2000, 679–689.

- [Mobasher 2000] Mobasher B., Cooley R., Srivastava J., *Automatic personalization based on Web usage mining*. Comm of the ACM 43(8), 2000, 142–151.
- [Montaner 2003] Montaner M., Lopez B., De La Rosa J.L., *A Taxonomy fo Recommender Agents on the Internet*. Artificial Intelligence Review, 19, 2003, 285–330.
- [Musiał 1989] Musiał K., Nguyen N.T., *On the nearest product of partitions*. Bull. of Polish Academy of Sci., 36(5–6), 1989, 333–338.
- [Nelson 1965] Nelson T., *Complex information processing: a file structure for the complex, the changing and the indeterminate*. [In:] Proc. of the 1965 20th national conference. Cleveland, Ohio, United States, 1965, 84 – 100.
- [Newman 1996] Newman W.M., Lamming M.G., *Interactive System Design*. Addison-Wesley: Harlow 1996.
- [Nguyen 2000] Nguyen N.T., *Using consensus methods for solving conflicts of data in distributed systems*. Lecture Notes of Computer Science 1963, 2000, 409–417.
- [Nguyen 2000a] Nguyen N.T., Sobecki J., *Uzgadnianie profilu obsługi użytkownika w modelu rozproszonych interfejsów z wykorzystaniem metod consensusu*. [W:] *Multimedialne i sieciowe systemy informacyjne 2000*. Materiały konferencyjne pod red. Czesława Daniłowicza. Oficyna Wydaw. PWr., Wrocław 2000, 121–127.
- [Nguyen 2001a] Nguyen N.T., *Using distance functions to solving representation choice problem*. Fundamenta Informaticae 48(4), 2001, 295–314.
- [Nguyen 2001b] Nguyen N.T., Sobecki J., *Consensus-based methods applied to the intelligent user interface development*. [In:] Proc. of Information System Modelling. Hradec nad Moravici, Czech Republic, May 9–11, 2001, 139–146.
- [Nguyen 2002] Nguyen N.T., Sobecki J., *Consensus problem of user classification in interactive systems*. [In:] Grzegorzewski P., Hryniewicz O., Gil M.A., *Soft methods in probability, statistics and data analysis*. Physica-Verlag, cop.: Heidelberg, New York 2002, 346–354.
- [Nguyen 2002a] Nguyen N.T., *Metody wyboru consensusu i ich zastosowanie w rozwiązywaniu konfliktów w systemach rozproszonych*. Oficyna Wydaw. PWr., Wrocław 2002.
- [Nguyen 2003] Nguyen N.T., Sobecki J., *Consensus versus Conflicts – Methodology and Applications*. Lecture Notes in Artificial Intelligence 2639, 2003, 565–572.
- [Nguyen 2003a] Nguyen N.T., Sobecki J., *Using Consensus Methods to Construct Adaptive Interfaces in Multimodal Web-based Systems*. Universal Access in Information Society, 2(4), 2003, 342 – 358.
- [Nguyen 2005] Nguyen N.T., Sobecki J., *Rough classification used for learning scenario determination in intelligent learning system*. [In:] Kłopotek M.A., Wierzchoń S., Trojanowski K. (eds.), *Intelligent information processing and web mining*. Springer, Berlin 2005, 107–116.
- [Nguyen 2006] Nguyen N.T., Sobecki J., *Determination of User Interfaces in Adaptive Systems Using a Rough Classification based Method*. New Generation Computing 24(4), 2006, 377–402.
- [Nguyen 2007] Nguyen N.T., Rakowski M., Rusin M., Sobecki J., Jain L.C., *Hybryd Filtering Methods Applied in Web-Based Movie Recommendation System*. Lecture Notes in Artificial Intelligence 4692, 2007, 206–213.
- [Nguyen 2008] Nguyen N.T., *Advanced Methods for Inconsistent Knowledge Management*. Springer Verlag, London 2008.
- [Nielsen 1995] Nielsen J., *Multimedia and Hypertext*. Academic Press Professional, Boston 1995.
- [Nielsen 2003] Nielsen J., *Projektowanie funkcjonalnych serwisów internetowych*. HELION, Gliwice 2003.

- [Nilson 1990] Nilson U., Maluszyński J., *Logic. Programming and Prolog*. John Wiley & Sons, Chichester 1990. <http://www.ida.liu.se/~ulfni/lpp/copyright.html>
- [Nystrand 2001] Nystrand J., *Multimodal noncommand user interfaces*. Lund University Cognitive Science, 2001.
- [O'Connell 1995] O'Connell M., *Java: The inside story*. SunWorld July 1995. Pobrano lipiec 2007 <http://sunsite.uakom.sk/sunworldonline/swol-07-1995/swol-07-java.html>.
- [Oh 2007] Oh K.J., Kim T.Y., *Financial market monitoring by case-based reasoning*. Expert Systems With Applications, 32(3), 2007, 789–800.
- [O'Reilly 2005] O'Reilly T., *What Is Web 2.0. Design Patterns and Business Models for the Next Generation of Software*. 09/30/2005. Pobrano lipiec 2007 <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html>
- [Papagelis 2005] Papagelis M., Plexousakis D., *Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents*. Engineering Applications of Artificial Intelligence 18, 2005, 781–789.
- [Patel-Schneider 2004] Patel-Schneider P.F., Hayes P., Horrocks I., *OWL Web Ontology Language Semantics and Abstract Syntax*, W3C Recommendation 2004. Pobrano lipiec 2007 <http://www.w3.org/TR/owl-semantics/>.
- [Pawlak 1983] Pawlak Z., *Systemy informacyjne. Podstawy teoretyczne*. WNT, Warszawa 1983.
- [Pawlak 1999] Pawlak Z., *Rough classification*. Int. J. Human-Computer Studies, 51, 1999, 369–383.
- [Pazzani 1996] Pazzani M., Muramatsu J., Billsus D., *Syskill & Webert: Identifying Interesting Web Sites*. [In:] Proc. of AAAI'96 / IAAI'96, Portland, OR. Menlo Park, CA, AAAI Press, 1996, 54–61.
- [Pearrow 2002] Pearrow M., *Funkcjonalność stron internetowych*. HELION, Gliwice 2002.
- [Pearson's 2006] *Pearson's correlation coefficient*. 17.01.2006. Pobrano styczeń 2008 [http://www.vias.org/tmdatanaleng/cc\\_corr\\_coeff.html](http://www.vias.org/tmdatanaleng/cc_corr_coeff.html).
- [Pearson's 2007] *Pearson's correlation coefficient*. 2007. Pobrano styczeń <http://www.texasoft.com/winkpear.html>.
- [Pieczyńska 2000] Pieczyńska-Kuchtiak A., Sobecki J., *Rozpoznawanie osób na podstawie obrazów cyfrowych zawierających ich twarze*. [W:] Multimedialne i sieciowe systemy informacyjne 2000. Wrocław, 18–19 września 2000. Oficyna Wydaw. PWr., Wrocław 2000, 207–216.
- [Pluta 1986] Pluta W., *Wielowymiarowa analiza porównawcza w modelowaniu ekonometrycznym*, PWN, Warszawa 1986.
- [Płoski 2001] Płoski Z., *Słownik Encyklopedyczny – Informatyka*. Wydawnictwa Europa 2001.
- [Preece 1996] Preece J., Rogers Y., Benyon D., Holland S., Carem T., *Human-computer interaction*. Addison-Wesley, Harlow 1996.
- [Quinlan 1986] Quinlan J.R., *Induction of decision trees*. Machine Learning 1, 1986, 81–106.
- [Quinn 2001] Quinn L.M., Pawasarat J., *Confronting Anti-Urban Marketing Stereotypes*. A Milwaukee Economic Development Challenge. June 2001. <http://www.uwm.edu/Dept/ETI/purchasing/markets.htm>.
- [Rada 1991] Rada R., *Hypertext: From Text to Expertext*, McGraw Hill Publishers 1991.
- [Rafaeli 1988] Rafaeli S., *Interactivity: From new media to communication*. Sage annual review of communication research: Advancing communication science 16, 1988, 110–134.

- [Rakowski 2006] Rakowski M., Rusin M., Sobecki J., *Hybrid recommendation applied in web-based movie information system*. [In:] Proc. of Multimedia and network information systems 2006. Wrocław, September 21–22, 2006, 361–369.
- [Raskin 2000] Raskin J., *The Human Interface. New Directions for Designing Interactive Systems*. Addison-Wesley, Boston 2000.
- [Riccardi 2001] Riccardi G., *Principles of Database Systems with Internet and Java Applications*. Addison Wesley, Boston 2001.
- [Rich 1979] Rich E., *User Modeling via Stereotypes*. Cognitive Science, 3, 1979, 329–354.
- [Rijsbergen 1979] Rijsbergen C., *Information Retrieval*. 2nd edition. Butterworths, London 1979.
- [RSI 2004] RSI Internet, *Personalizacja*. Pobrano maj 2004 <http://www.webbox.pl/personalizacja.php>.
- [Rudowski 2005] Rudowski M., *Analiza efektywności aplikacjiweb-owych Flash MX*. Praca magisterska Wydział Informatyki i Zarządzania Politechniki Wrocławskiej. Promotor J. Sobecki. Wrocław 2005.
- [Salton 1975] Salton G. (red.), *SMART – automatyczny system wyszukiwania informacji*. WNT, Warszawa 1975.
- [Salton 1983] Salton G., McGill M.J., *Introduction To Modern Information Retrieval*, McGraw-Hill Book Co., New York 1983.
- [Sarwar 2001] Badrul M., Sarwar G., Karypis J.A., Konstan J., Reidl, *Item-based collaborative filtering recommendation algorithms*. [In:] *World Wide Web*, 2001, 285–295.
- [Schneiderman 1997] Schneiderman B., Maes P., *Debate: Direct manipulation vs. interface agents*. Interactions 4(6), 1997, 42–61.
- [Schwab 2000] Schwab I., Kobsa A., Koychev I., *Learning about Users from Observations*. [In:] Proc. Adaptive User Interfaces the 2000 AAAI Spring Symposium. Menlo Park, CA, AAAI Press 2000, 102–106.
- [Setten 2005] van Setten M., *Supporting People in Finding Information. Hybrid Recommender Systems And Goal-Based Structuring*. Enschede, The Netherlands, 2005, Telematica Instituut Fundamental Research Series, No. 016.
- [Shannon 2006] Shannon V., *A more revolutionary Web*. International Herald Tribune (2006-06-26). Pobrano lipiec 2007 <http://www.ihf.com/articles/2006/05/23/business/web.php>.
- [Shardanand 1995] Shardanand U., Maes P., *Social Information Filtering: Algorithms for Automating Word of Mouth*. [In:] Proc. of ACM CHI'95 Conference on Human Factors in Computing Systems. ACM, New York, 1995, 210–217. [http://citeseer.ist.psu.edu/cache/papers/cs/15862/http:SzzSzmaz.cs.umass.eduSz~aseltinezSz791SzSzshardanand.social\\_information\\_filtering.pdf/shardanand95social.pdf](http://citeseer.ist.psu.edu/cache/papers/cs/15862/http:SzzSzmaz.cs.umass.eduSz~aseltinezSz791SzSzshardanand.social_information_filtering.pdf/shardanand95social.pdf)
- [Shearin 2001] Shearin S., Lieberman H., *Intelligent Profiling by Example*. [In:] Proc of the Conf on Intelligent User Interfaces, ACM Press, 2001.
- [Skarbek 1998] Skarbek W. (red.), *Multimedia – Algorytmy i standardy kompresji*, AOW PLJ, Warszawa 1998.
- [Smith 2004] Smith R.M., *FAQ: Web Bugs*. Pobrano czerwiec 2004 <http://www.privacyfoundation.org/resources/webbug.asp>.
- [Sobecki 1989] Sobecki J., Trzcionka T., *Mikrokomputerowy system analizy morfologicznej MOR-FAN*. Raporty Bibl. Gł. OINT PWr 1989 Ser. PRE nr 125.

- [Sobecki 1995] Sobecki J., *The concept of the WWW support scientific information exchange*. [In:] Proc. of 17<sup>th</sup> Informations Systems Architecture and Technology ISAT '95, Szklarska Poręba, 25–30 września 1995, 197–202.
- [Sobecki 1997] Sobecki J., Piotrowska J., *Ewidencja i rozpowszechnianie dorobku naukowego z wykorzystaniem usług informacyjnych sieci Internet*. [W:] *Bibliotekarze i użytkownicy bibliotek w dobie powszechnej komputeryzacji (wzajemne relacje, potrzeby, związki)*. Materiały konferencyjne. Pod red. Jolanty Goc. Biblioteka Główna Uniw. Szczec., [Biblioteka Główna] PSzczec. Szczecin, 13–15 listopada 1996. Szczecin, Wydaw. Nauk. Uniw. Szczec. 1997, 202–220.
- [Sobecki 1997a] Sobecki J., Piotrowska J., *Technical reports Internet dissemination*. [In:] Proc. of 5th international conference on Systems integration '97. 1997, 495–503.
- [Sobecki 1998] Sobecki J., *Analiza możliwości wyszukiwania obiektów multimedialnych w systemie WWW*. [W:] *Multimedialne i sieciowe systemy informacyjne 1998*. Materiały konferencyjne pod red. Czesława Daniłowicza. [Wrocław, 17–18 września 1998]. Wrocław, Oficyna Wydaw. PWr. 1998, 189–199.
- [Sobecki 1999] Sobecki J., Domańska A., *Projektowanie środowisk nauczania na odległość pod WWW na przykładzie kursu programowania logicznego*. [W:] *Rola edukacji informatycznej w technice i jej związek z wychowaniem ekologicznym i kształceniem zawodowym*. V Ogólnopolskie Warsztaty Multimedialne w Dydaktyce Techniki. Pod red. T. Brodzińskiego. Szczecin, 19–20 listopada 1998. Szczecin, Wydaw. Nauk. Uniw. Szczec. 1999, 91–100.
- [Sobecki 1999a] Sobecki J., *Multimedia interactive system evaluation*. [In:] Proc. of the 21<sup>st</sup> International Conference on Information Systems Architecture and Technology ISAT '99. Łądek Zdrój, 22–25 September 1999. Wrocław, Oficyna Wydaw. PWroc. 1999, 241–245.
- [Sobecki 1999b] Sobecki J., *Rozpowszechnianie informacji naukowych w postaci cyfrowej*. [W:] VI Krajowa konferencja KOWBAN '99 Komputerowe wspomaganie badań naukowych. Materiały. Wrocławskie Towarzystwo Naukowe. Wrocław–Polanica Zdrój, 28–30 października 1999. Wrocław, [Oficyna Wydaw. Sudety] 1999, 423–428.
- [Sobecki 2000] Sobecki J., *Interactive multimedia information systems planning*. [W:] *Digital media in networks 1999. Perspectives to digital world*. Ed. Timo Valiharju. Tampere, University of Tampere. Computer Center/Hypermedia Laboratory 2000, 38–44.
- [Sobecki 2000a] Sobecki J., Nguyen N.T., *Towards distributed learning of logic: a design perspective*. [In:] Proc. of 2nd Global Congress on Engineering Education. Wismar, 2–7 July 2000, 162–164.
- [Sobecki 2001] Sobecki J., Nguyen N.T., *Consensus-based adaptive user interface for universal access systems*. [In:] Proc. 9th International Conference on Human-Computer Interaction. Symposium on Human Interface. 4th International Conference on Engineering Psychology and Cognitive Ergonomics. 1st International Conference on Universal Access in Human-Computer Interaction. New Orleans, Louisiana, USA, August 5–10, 2001. Vol. 3, 112–116.
- [Sobecki 2001a] Sobecki J., Nguyen N.T., *Adaptive user interfaces for general access web systems*. [In:] Proc. of the 14th International Conference on Systems Science. Eds. Z. Bubnicki, A. Grzech. Wrocław, 11–14 September 2001. Vol. 3. Information systems. Technical applications. Non-technical applications. Wrocław, Oficyna Wydaw. PWroc. 2001, 70–77.
- [Sobecki 2001b] Sobecki J., *One suits all – is it possible to build a single interface appropriate for all users?* [In:] Grzech A., Wilimowska Z. (eds.), *Proceedings of the 23rd Int. Scientific School ISAT*, PWr Press Wrocław, 2001, 125–131.

- [Sobecki 2002] Sobecki J., *Adaptive interface for hypermedia applications*. [In:] Proc. of VIProm-Com-2002 – 4th EURASIP – IEEE Region 8 International Symposium on Video Processing and Multimedia Communications, Zadar, Croatia, 16–19 June 2002, 147–152.
- [Sobecki 2002a] Sobecki J., *Socially adaptive user interfaces for web-based systems*. [In:] Proc. Of the 7<sup>th</sup> ERCIM Workshop “User Interfaces for All” Chantilly France, 24025 October 2002, 67–76.
- [Sobecki 2002b] Sobecki J., *Interface Model in Adaptive Web-based System*. [W:] Cellary W., Iyengar A. (eds.), Internet Technologies, Applications and Societal Impact. Kluwer Academic Publishers, Boston 2002, 93–104.
- [Sobecki 2003] Sobecki J., *XML-based Interface Model for Socially Adaptive Web-Based Systems User Interfaces*. Lecture Notes in Computer Science 2660, 2003, 592–598.
- [Sobecki 2003a] Sobecki J., Nguyen N.T., *Consensus-based Adaptive Interface Construction for Multiplatform Web Applications*. Lecture Notes in Computer Science, 2690, 2003, 457–461.
- [Sobecki 2003b] Sobecki J., Morel K., Bednarczuk T., *Web-based intelligent tutoring system with strategy selection using consensus methods*. [In:] Proc. of International Conference on Intelligent information processing and Web Mining IIS:IIPWM’03. M.A. Kłopotek, S.T. Wierzchoń, K. Trojanowski (eds.). Zakopane, June 2–5, 2003. Berlin [i in.], Springer, cop. 2003, 105–109.
- [Sobecki 2004] Sobecki J., *Hybrid Adaptation of Web-Based Systems User Interfaces*. Lecture Notes in Computer Science 3038, 2004, 505–512.
- [Sobecki 2004a] Sobecki J., *Web-based systems user interfaces recommendation using hybrid methods*. [In:] Intelligent technologies for inconsistent knowledge processing. Ed. Ngoc Thanh Nguyen. Magill, Advanced Knowledge International, cop. 2004, 95–111.
- [Sobecki 2004b] Sobecki J., Weihberg M., *Consensus-based Adaptive User Interface Implementation in the Product Promotion*. [In:] Design for a more inclusive world, Keates S., Clarkson J., Langdon P. and Robonson P., Springer-Verlag, London 2004, 111–122.
- [Sobecki 2005] Sobecki J., *Consensus-based hybrid adaptation of web systems user interfaces*. Journal of Universal Computer Science, 11(2), 2005, 250–270.
- [Sobecki 2006] Sobecki J., *Implementations of Web-based Recommender Systems Using Hybrid Methods*. International Journal of Computer Science & Applications, 3(3), 2006, 52–64. <http://www.tmrfindia.org/ijcsa/V3I35.pdf>.
- [Sobecki 2006a] Sobecki J., Babiak E., Ślanina M., *Application of hybrid recommendation in web-based cooking assistant*. Lecture Notes in Artificial Intelligence 4253, 2006, 797–804.
- [Sobecki 2007] Sobecki J., Szczepański L., *Wiki-news interface agent based on AIS methods*. Lecture Notes in Artificial Intelligence. 4496, 2007, 258–266.
- [Sobecki 2007a] Sobecki J., *Web-based system user interface hybrid recommendation using ant colony metaphor*. Lecture Notes in Artificial Intelligence. 4694, 2007, 1033–1040.
- [Sobecki 2007b] Sobecki J., Żatuchin D., *Badanie użyteczności aplikacji webowych z wykorzystaniem urządzeń do śledzenia wzroku*. IV Konferencja Jakość Systemów Informatycznych, Jachranka 22–23 maja 2007.
- [Sobecki 2008] Sobecki J., *Image identification based on the pupil size analysis during human-computer interaction*. [W:] New trends in multimedia and network information systems. Zgrzywa A., Choroś K., Siemiński A. (eds.). IOS Press Amsterdam 2008, 16–22.
- [Sobecki 2008a] Sobecki J., *Ant colony metaphor applied in user interface recommendation*. New Generation Computing. 26(3), 2008, 277–293.



- [Sobecki 2008b] Sobecki J., *Hybrydowe systemy rekomendacyjne zastosowane w web-owych systemach interakcyjnych*. [W:] Rozwój informatycznych systemów wieloagentowych w środowiskach społeczno-gospodarczych. Red. nauk. Stanisław Stanek [i in.]. Warszawa, Wydaw. PLA-CET, 2008, 151–167.
- [Sobecki 2009] Sobecki J., Piwowar K., *Verification of recommendation methods*. [In:] e-commerce applications. First Asian Conf. On Intelligent Information and Database Systems. IEEE 2009, 127–131.
- [Soergel 1967] Soergel D., *Mathematical analysis of documentation systems: An attempt to a theory of classification and search request formulation*. Information Storage and Retrieval 3(3), 1967, 129–173.
- [Spolsky 2001] Spolsky J., *Projektowanie interfejsu użytkownika. Poradnik programistów*. MIKOM, Warszawa 2001.
- [Stary 2001] Stary C., *User diversity and design representation: Towards increased effectiveness in Design for All*. Int J UAIS 1, 2001, 16–30.
- [Steinbrink 1993] Steinbrink B., *Multimedia u progu technologii XXI w.* Wyd. Robomatic, Wrocław 1993.
- [Steinhaus 1956] Steinhaus H., *Sur la division des corps matériels en parties*. Bull. Acad. Polon. Sci 4, 1956, 801–804. Przedruk w: Steinhaus H., Selected Papers. PWN, Warszawa 1985.
- [Stephanidis 1998] Stephanidis C., Paramythis A., Akoumianakis D., Sfyrakis M., *Self-adapting Web-based systems: towards universal accessibility*. [In:] Proc. ERCIM Workshop 1998.
- [Stephanidis 2001] Stephanidis C., Savidis A., *Universal Access in the Information Society: Methods, Tools, and Interaction Technologies*. Int. J. UAIS, 1, 2001, 40–55.
- [Strykowski 1984] Strykowski W., *Audiowizualne materiały dydaktyczne*. PWN, Warszawa 1984.
- [Sugeno 1985] Sugeno M., *An introductory survey of fuzzy control*. Information Systems 36, 1985 59–83.
- [Symeonidis 2007] Symeonidis T., Nanopoulos A., Manolopoulos Y., *Feature-weighted User Model for Recommender Systems*. Proc. of UM 2007.
- [Tanaka 2005] Tanaka K., Fukui K., *A user interface via agent-based interaction using gaze input*. Electronics and Communications in Japan (Part II: Electronics), 88 (3), 2005, 44–52.
- [Taylor 1995] Taylor A., *Mathematics and Politics: Strategy, Voting, Power, and Proof*. New York, Springer-Verlag, 1995.
- [Taylor 2002] Taylor Nelson Sofres Interactive, *Global eCommerce Report 2002*.
- [Tidwell 2005] Tidwell J., *Designing Interfaces*. O'Reilly 2005. <http://designinterfaces.com>.
- [Timmis 2004] Timmis J., Knight T., De Castro L.N., Hart E., *An Overview of Artificial Immune Systems*. [In:] R. Paton, H. Bolouri, M. Holcombe, J.H. Parish, and R. Tateson, editors, *Computation in Cells and Tissues: Perspectives and Tools for Thought*, Natural Computation Series, 51–86. Springer, November 2004.
- [Ullmer 2005] Ullmer B., Ishii H., Jacob R.J.K., *Token+Constraint Systems for Tangible Interaction with Digital Information*, ACM Transactions on Computer-Human Interaction, 12(1), 2005, 81–118
- [Underhal 2002] Underhal B., *Głębsze spojrzenie. Flash MX*. Helion, Gliwice 2002.
- [Underhal 2002a] Underhal B., *Język programowania ActionScript. Flash MX*. Helion, Gliwice 2002.

- [URG 2002] User Centered Design Methods, *Usability Consulting Services*, Indiana University Acknowledgement 2002. Pobrano sierpień 2005 [http://www.indiana.edu/~usable/presentations/ucd\\_methods.pdf](http://www.indiana.edu/~usable/presentations/ucd_methods.pdf)
- [UsabilityNet 2006] UsabilityNet, *Methods Table*. UsabilityNet 2006. Pobrano wrzesień 2007 <http://www.usabilitynet.org/tools/methods.htm>.
- [Viera 2005] Viera A.J., Garrett J.M., *Understanding Interobserver Agreement: The Kappa Statistic*, *Family Medicine* 37(5), 2005, 360–363.
- [Voelkel 2006] Voelkel M., *Usability Tests and Eye Tracking*. SCOREBERLIN Usability tests and eyetracking GMBH. 05/2006. Pobrano lipiec 2007 <http://www.scoreberlin.de/usability-artikel/usability-tests-eyetracking-en/>.
- [Vrandecic 2007] Vrandecic D., Krötzsch M., Völkel M., *Die Zukunft der Wikis: Semantic Web*. [W:] Christoph Lange, Wikis und Blogs: Planen, Einrichten, Verwalten, C&L Computer und Literaturverlag, chapter 15, January 2007, 789–810.
- [W3C 1996] W3C, *Extensible Markup Language. W3C Working Draft 961114*. W3C 1996. Pobrano grudzień 2008 <http://www.w3.org/TR/WD-xml-961114.html>.
- [W3C 2003] W3C, *Some early ideas for HTML*. W3C 2003. Pobrano czerwiec 2006 <http://www.w3c.org/MarkUp/historical>.
- [Waglowski 2004] Waglowski P., Majewski W., *Prosto, przejrzyste i wygodnie*. [W:] Raport specjalny Computerworld: IT w sektorze publicznym, kwiecień 2004, 16. Pobrano lipiec 2005 <http://www.isoc.org.pl/wiki/index.php/WebAccessibility/DlaPolski>.
- [Waldron 2006] Waldron R., *The Flash History*. Flashmagazine.com Published: 31 Jul 2002 Updated: 27 Aug 2006. Pobrano lipiec 2007 <http://www.flashmagazine.com/413.htm>.
- [Ward 2008] Ward C., *Wiki History*. Pobrano styczeń 2009 <http://c2.com/cgi/wiki?WikiHistory>.
- [Welling 2003] Welling L., Thomson L., *PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty*. Wydanie 2. Helion, Gliwice 2003.
- [Whalen 2002] Whalen D., 'The Unofficial Cookie FAQ, Version 2.54'. Contributed to Cookie Central by David Whalen. Pobrano lipiec 2003 <http://www.cookiecentral.com/faq/#2.7>.
- [Wickelgreen 2004] Wickelgreen I., *Brain-Computer Interface a New Dimension*, *Science* 206, 2004, 1878–1879.
- [Wilson 2000] Wilson R.J., *Wprowadzenie do teorii grafów*. PWN, Warszawa 2000.
- [Witkowski 2002] Witkowski M., *Po drugiej stronie*. Chip 5, 2002. Pobrano lipiec 2007 [http://www.chip.pl/arts/archiwum/n/articlear\\_23881.html](http://www.chip.pl/arts/archiwum/n/articlear_23881.html)
- [Wooldridge 1995] Wooldridge M., Jennings N.R., *Intelligent agents: theory & practice*. Knowledge Engineering Review 100(2), 1995, 115–152.
- [Wu 2005] Wu G., *The Mediating Role of Perceived Interactivity in the Effect of Actual Interactivity on Attitude toward the Website*, *Journal of Interactive Advertising*, 5(2), 2005.
- [XUL 2007] *The Joy of XUL*. Mozilla developer center. Pobrano lipiec 2007 [http://developer.mozilla.org/en/docs/The\\_Joy\\_of\\_XUL](http://developer.mozilla.org/en/docs/The_Joy_of_XUL).
- [Zieliński 2000] Zieliński J.S. (red.), *Inteligentne systemy w Zarządzaniu. Teoria i praktyka*. PWN, Warszawa 2000.
- [Żatuchin 2008] Żatuchin D., *Ślady użytkowników w procesie badania użyteczności stron WWW*. Materiały III Seminarium Interfejs Użytkownika KANSEI w Praktyce, Warszawa 2008, PJWSTK.





- [URG 2002] User Centered Design Methods, *Usability Consulting Services*, Indiana University Acknowledgement 2002. Pobrano sierpień 2005 [http://www.indiana.edu/~usable/presentations/ucd\\_methods.pdf](http://www.indiana.edu/~usable/presentations/ucd_methods.pdf)
- [UsabilityNet 2006] UsabilityNet, *Methods Table*. UsabilityNet 2006. Pobrano wrzesień 2007 <http://www.usabilitynet.org/tools/methods.htm>.
- [Viera 2005] Viera A.J., Garrett J.M., *Understanding Interobserver Agreement: The Kappa Statistic*, Family Medicine 37(5), 2005, 360–363.
- [Voelkel 2006] Voelkel M., *Usability Tests and Eye Tracking*. SCOREBERLIN Usability tests and eyetracking GMBH. 05/2006. Pobrano lipiec 2007 <http://www.scoreberlin.de/usability-artikel/usability-tests-eyetracking-en/>.
- [Vrandecic 2007] Vrandecic D., Krötzsch M., Völkel M., *Die Zukunft der Wikis: Semantic Web*. [W:] Christoph Lange, Wikis und Blogs: Planen, Einrichten, Verwalten, C&L Computer und Literaturverlag, chapter 15, January 2007, 789–810.
- [W3C 1996] W3C, *Extensible Markup Language. W3C Working Draft 961114*. W3C 1996. Pobrano grudzień 2008 <http://www.w3.org/TR/WD-xml-961114.html>.
- [W3C 2003] W3C, *Some early ideas for HTML*. W3C 2003. Pobrano czerwiec 2006 <http://www.w3c.org/MarkUp/historical>.
- [Wagłowski 2004] Wagłowski P., Majewski W., *Prosto, przejrzyste i wygodnie*. [W:] Raport specjalny Computerworld: IT w sektorze publicznym, kwiecień 2004, 16. Pobrano lipiec 2005 <http://www.isoc.org.pl/wiki/index.php/WebAccessibility/DlaPolski>.
- [Waldron 2006] Waldron R., *The Flash History*. Flashmagazine.com Published: 31 Jul 2002 Updated: 27 Aug 2006. Pobrano lipiec 2007 <http://www.flashmagazine.com/413.htm>.
- [Ward 2008] Ward C., *Wiki History*. Pobrano styczeń 2009 <http://c2.com/cgi/wiki?WikiHistory>.
- [Welling 2003] Welling L., Thomson L., *PHP i MySQL. Tworzenie stron WWW. Vademecum profesjonalisty*. Wydanie 2. Helion, Gliwice 2003.
- [Whalen 2002] Whalen D., 'The Unofficial Cookie FAQ, Version 2.54'. Contributed to Cookie Central by David Whalen. Pobrano lipiec 2003 <http://www.cookiecentral.com/faq/#2.7>.
- [Wickelgreen 2004] Wickelgreen I., *Brain-Computer Interface a New Dimension*, Science 206, 2004, 1878–1879.
- [Wilson 2000] Wilson R.J., *Wprowadzenie do teorii grafów*. PWN, Warszawa 2000.
- [Witkowski 2002] Witkowski M., *Po drugiej stronie*. Chip 5, 2002. Pobrano lipiec 2007 [http://www.chip.pl/arts/archiwum/n/articlear\\_23881.html](http://www.chip.pl/arts/archiwum/n/articlear_23881.html)
- [Wooldridge 1995] Wooldridge M., Jennings N.R., *Intelligent agents: theory & practice*. Knowledge Engineering Review 100(2), 1995, 115–152.
- [Wu 2005] Wu G., *The Mediating Role of Perceived Interactivity in the Effect of Actual Interactivity on Attitude toward the Website*, Journal of Interactive Advertising, 5(2), 2005.
- [XUL 2007] *The Joy of XUL*. Mozilla developer center. Pobrano lipiec 2007 [http://developer.mozilla.org/en/docs/The\\_Joy\\_of\\_XUL](http://developer.mozilla.org/en/docs/The_Joy_of_XUL).
- [Zieliński 2000] Zieliński J.S. (red.), *Inteligentne systemy w Zarządzaniu. Teoria i praktyka*. PWN, Warszawa 2000.
- [Żatuchin 2008] Żatuchin D., *Ślady użytkowników w procesie badania użyteczności stron WWW*. Materiały III Seminarium Interfejs Użytkownika KANSEI w Praktyce, Warszawa 2008, PJWSTK.







Liczba użytkowników systemów webowych zastosowania stale wzrasta. Powoduje to rosnącą liczbę użytkowników, jak i stosowanych platform systemowo-sprzętowych. Zaprojektowanie interfejsu użytkownika systemu webowego, jednakowego dla wszystkich potencjalnych użytkowników tego systemu i wystarczająco użytecznego dla wszystkich z nich, jest praktycznie niemożliwe. Metody projektowania interfejsów użytkownika w systemach komputerowych, opracowane w ramach dziedziny HCI, zalecają zaprojektować interfejs dostosowany do celu aplikacji, podstawowej grupy użytkowników oraz platformy systemowo-sprzętowej. Współczesne systemy webowe coraz częściej są wyposażone w mechanizmy umożliwiające adaptację interfejsu do wszystkich lub choćby części z wymienionych czynników. Systemy tego typu nazywa się spersonalizowanymi lub adaptacyjnymi. W pracy przedstawiono krótką historię i rozwój systemów webowych, metodologię projektowania funkcjonalnych serwisów webowych, modelowanie użytkowników w systemach webowych, rekomendację interfejsu w systemach webowych z zastosowaniem wybranych metod demograficznych, kolaboratywnych i hybrydowych oraz metody wyznaczania profilu użytkownika z wykorzystaniem klasyfikacji przybliżonej.



Wydawnictwa Politechniki Wrocławskiej  
są do nabycia w księgarni „Tech”  
plac Grunwaldzki 13, 50-377 Wrocław  
budynek D-1 PWr., tel 071 320 29 35  
Prowadzimy sprzedaż wysyłkową

ISBN 978-83-7493-465-7