

WOJSKOWA AKADEMIA TECHNICZNA

BIULETYN

INSTYTUTU SYSTEMÓW INFORMATYCZNYCH

BIULETYN INSTYTUTU SYSTEMÓW INFORMATYCZNYCH
WYDZIAŁ CYBERNETYKI
WOJSKOWA AKADEMIA TECHNICZNA

ISSN 1508-4183

00-908 Warszawa, ul. S. Kaliskiego 2
tel.: (22)683-95-04, fax: (22)683-78-58
<http://www.isi.wat.edu.pl>

Zespół redakcyjny w składzie:

prof. dr hab. inż. Marian Chudy (redaktor naczelny)
dr hab. inż. Andrzej Walczak, prof. WAT (z-ca redaktora naczelnego)
prof. dr hab. inż. Andrzej Ameljańczyk
dr hab. inż. Ryszard Antkiewicz, prof. WAT
dr hab. inż. Andrzej Najgebauer, prof. WAT
dr hab. inż. Tadeusz Nowicki, prof. WAT
dr hab. inż. Bolesław Szafrąński, prof. WAT
dr hab. inż. Kazimierz Worwa, prof. WAT
dr inż. Zbigniew Tarapata (sekretarz naukowy)

Redaktorzy numeru:

dr hab. inż. Tadeusz Nowicki, prof. WAT
dr hab. inż. Kazimierz Worwa, prof. WAT

Redakcja techniczna i projekt graficzny: Barbara Fedyna

Druk: BEL Studio Sp. z o.o., 01-355 Warszawa, ul. Powstańców Śląskich 67B

SPIS TREŚCI

1. *A. Ameljańczyk* – Analiza wpływu przyjętej koncepcji modelowania systemu wspomagania decyzji medycznych na sposób generowania ścieżek klinicznych 1
2. *A. Ameljańczyk* – Matematyczne aspekty modelowania pajęczynowego obiektów 11
3. *T. Górski* – Analiza przydatności wybranych standardów do modelowania architektury systemu informatycznego dla służby zdrowia” 19
4. *P. Kosiuczenko* – Przydatność języka Gello i standardu RIM do obiektowego modelowania systemów medycznych 31
5. *M. Mazurek* – Wielokryterialne zadanie optymalizacji schematu agregatów 37
6. *A. Najgebauer, R. Kasprzyk* – Symulator rozprzestrzeniania się złośliwego oprogramowania w sieciach komputerowych 43
7. *G. Sabak* – Metoda analizy ruchu drogowego na podstawie zachowania użytkowników sieci telefonii komórkowej 51
8. *J. Wiśniewska* – Określanie priorytetów zmiennych pewnych funkcji decyzyjnych 61

Analiza wpływu przyjętej koncepcji modelowania systemu wspomaganie decyzji medycznych na sposób generowania ścieżek klinicznych

A. AMELJAŃCZYK

e-mail: aameljańczyk@wat.edu.pl

Instytut Systemów Informatycznych
Wydział Cybernetyki WAT
ul. S. Kaliskiego 2, 00-908 Warszawa

W pracy dokonano analizy najczęściej stosowanych koncepcji modelowych w konstrukcji systemów wspomaganie decyzji medycznych pod kątem ich wpływu na sposób generowania ścieżek klinicznych w obszarze tzw. „węzłów decyzyjnych”. Analizie poddano takie koncepcje modelowania jak modelowanie bazujące na teorii zbiorów rozmytych, teorii zbiorów przybliżonych, sieci bayesowskich oraz koncepcji wzorców deterministycznych.

Słowa kluczowe: Clinical Decision Support System, Clinical Pathways, Fuzzy Sets, Rough Sets, Multicriteria Optimization.

1. Wprowadzenie

Komputerowe Systemy Wspomaganie Decyzji Medycznych (Klinicznych) w skrócie KSWDM rozumiane są jako systemy interaktywnych programów komputerowych działających na określonej platformie sprzętowo – sieciowej, które w wyniku przetwarzania rezultatów badań medycznych i wiedzy medycznej wspierają proces diagnozy lekarskiej oraz metod leczenia. Z kolei ścieżki kliniczne (Clinical Pathway – CP) definiuje się [13], [17] jako system powiązanych działań, zapisany w określonej notacji (np. BPMN, GELLO, UML, OCL, XML lub innej), służący wspieraniu lekarzy oraz niższego personelu medycznego na etapie diagnozowania, planowania sposobu leczenia i jego przebiegu, prowadzący do wzrostu jakości i efektywności usług medycznych. CP może zatem posiadać dowolną formę zapisu, niekoniecznie w postaci programu komputerowego, co jest szczególnie znamienne dla KSWDM. Nie mniej jednak każda ścieżka kliniczna dowolnie zapisana, oprócz sekwencji zadań i działań zawierać musi tzw. „węzły decyzyjne”, które są elementami sterującymi w procesie leczenia konkretnego pacjenta w diagnozowanej chorobie. Z punktu widzenia

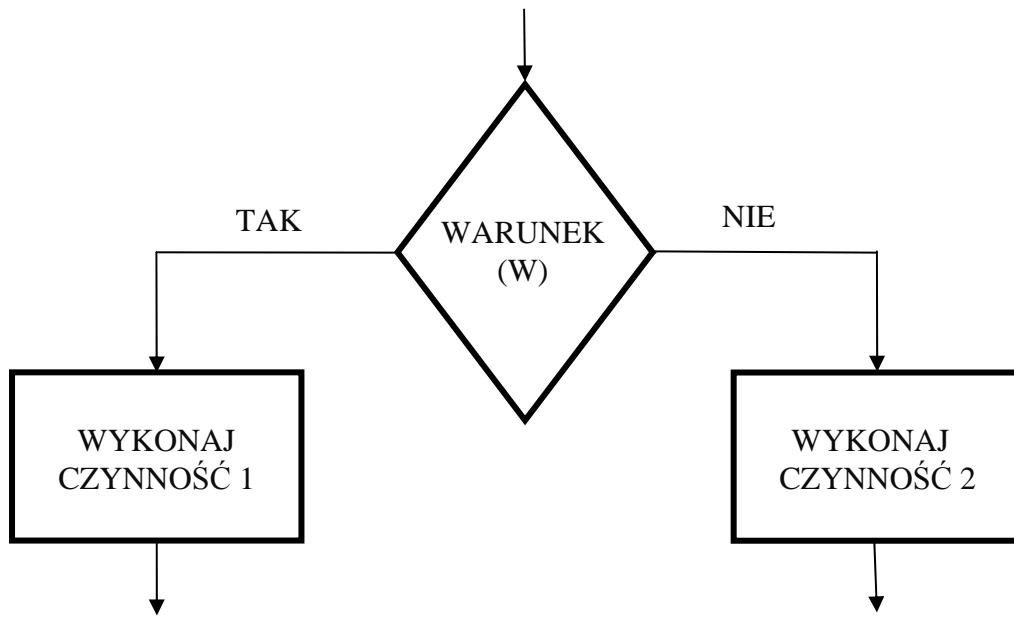
poprawności i skuteczności leczenia właśnie te elementy są w głównej mierze decydujące.

W zależności od stopnia „zinformalizowania”, rozstrzygnięcia w węzłach decyzyjnych ścieżek klinicznych odbywają się bądź to z udziałem wyłącznie lekarza (ewentualnie niższego personelu medycznego) bądź też dodatkowo ze wsparciem KSWDM.

Niniejsze opracowanie, zgodnie z założeniami projektu dotyczy tej drugiej sytuacji. Kompleksowy kształt (opis w określonym języku modelowania) ścieżki zależy zatem od przyjętej koncepcji modelowej zapisu danych medycznych oraz reguł wnioskowania KSWDM.

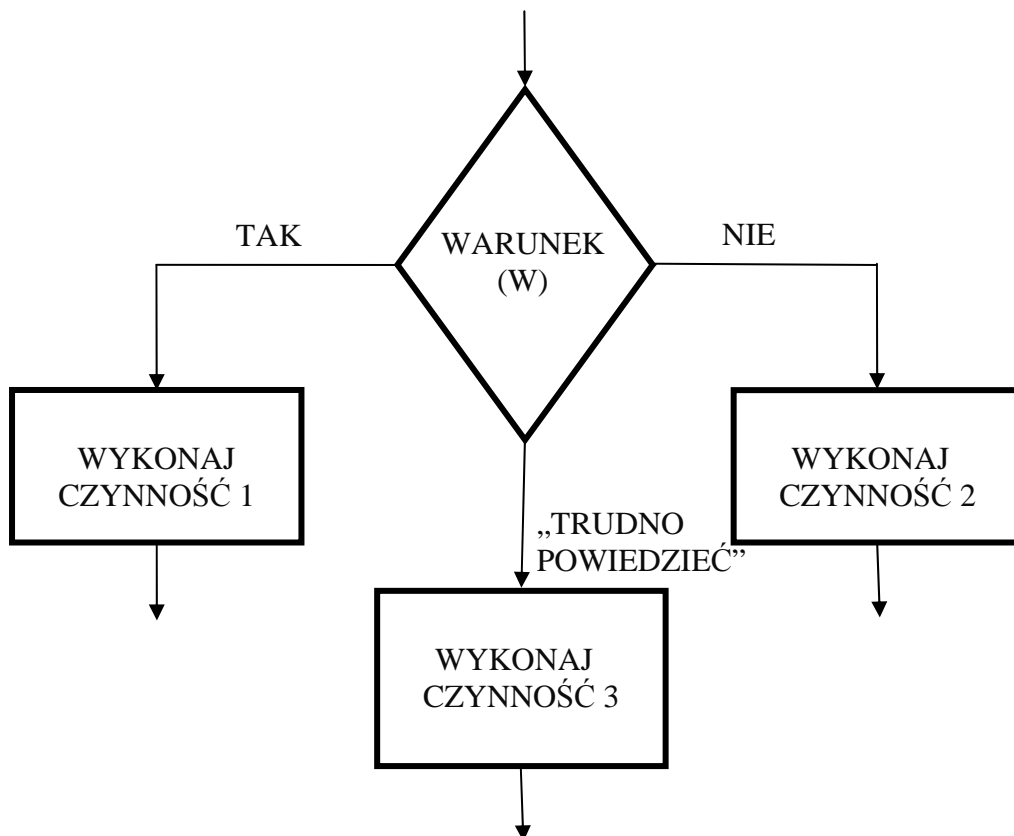
W dalszej części opracowania dokonana zostanie analiza najczęściej stosowanych języków modelowania matematycznego (koncepcji modelowych), służących modelowaniu i projektowaniu systemów wspomaganie decyzji medycznych pod kątem ich zalet i wad w kontekście aplikacyjnym generowanie ścieżek klinicznych.

Poniżej przedstawione zostały najczęściej występujące węzły decyzyjne w ścieżkach klinicznych:



Rys. 1. Prosty węzeł decyzyjny

lub bardziej skomplikowany:



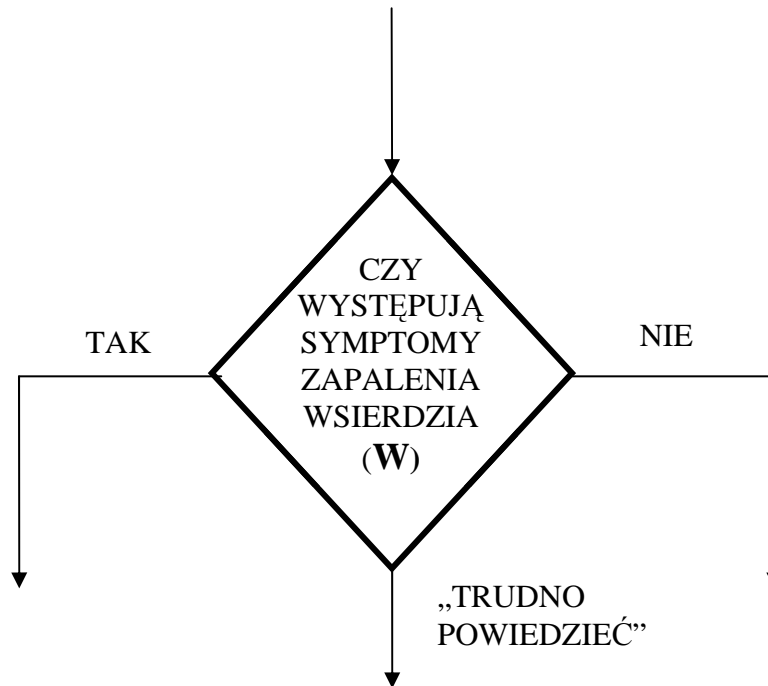
Rys. 2. Złożony węzeł decyzyjny

W większości przypadków spełnienie warunku W można sprowadzić do rozstrzygnięcia czy pewien element $w \in W$ czy też element $w \notin W$. Od wyniku tego rozstrzygnięcia zależy dalsza sekwencja działań zgodnie z możliwościami zapisanymi w ścieżce klinicznej. W przypadku ścieżki klinicznej realizowanej bez udziału KSWDM rozstrzygnięć tych w pełni dokonuje personel medyczny odpowiedniego szczebla, w przypadku zaś możliwości wsparcia decyzji medycznej przez system KSWDM rozstrzygnięć tych dokonuje również personel medyczny wzbogacony jednak „sugestią” tegoż systemu.

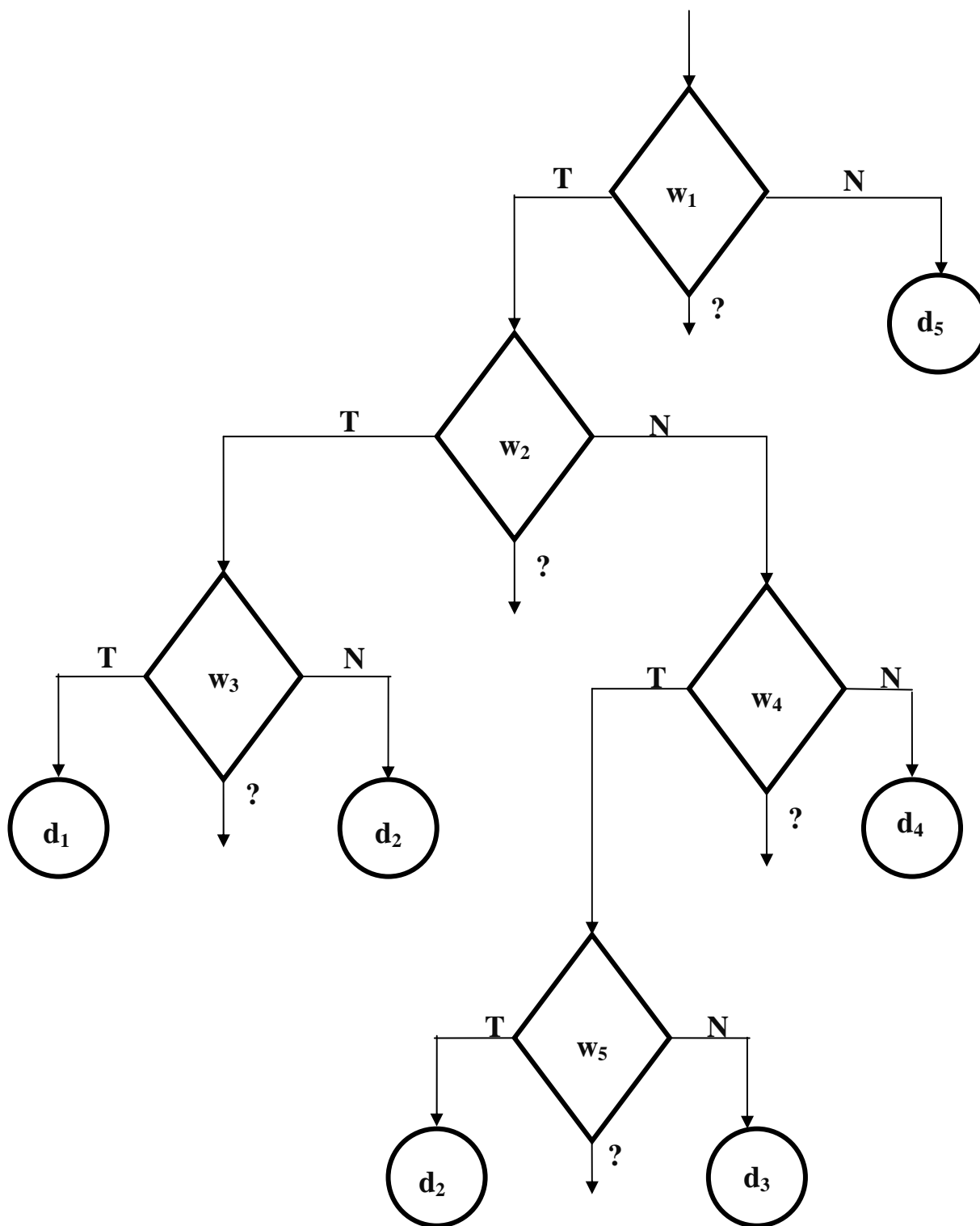
Poniższy schemat zapytania o spełnienie warunku W , według ekspertów, w wielkim uproszczeniu można uszczegółwić (rozbudować) następująco:

2. Koncepcja modelowania systemu wspomaganie decyzji medycznych oparta na teorii zbiorów rozmytych

Problem złożoności modelowania procesu wspomaganie decyzji diagnostycznych prześledźmy na uproszczonym przykładzie wstępnej diagnozy choroby serca jaką jest zapalenie wsierdzia.



Rys. 3. Diagnostyczny węzeł decyzyjny ścieżki klinicznej



Rys. 4. Częstkowe węzły decyzyjne węzła W

Możemy zatem zapisać:

$$W = \{w_1, w_2, w_3, w_4, w_5\}$$

gdzie:

w_1 = „czy występuje gorączka?”

w_2 = „czy występuje szybka akcja serca?”

w_3 = „czy występują silne dreszcze?”

w_4 = „czy występuje łatwe męczenie?”

w_5 = „czy występuje nadmierna potliwość?”

Zgodnie z sugestią specjalistów (ekspertów) w zależności od poszczególnych rozstrzygnięć (opcja „tak, nie” w węzłach decyzyjnych,

diagnozę w zakresie „zapalenia wsierdza” można sformułować jako jedną z następujących :

d_1 – bardzo duże ryzyko zapalenia wsierdza

d_2 – duże ryzyko

d_3 – średnie ryzyko

d_4 – małe ryzyko

d_5 – bardzo małe ryzyko

$$D = \{d_1, d_2, d_3, d_4, d_5\}$$

Już powierzchowna analiza rozstrzygnięć (opcja „tak, nie”) w zakresie cząstkowych warunków (symptomów) w_1, \dots, w_5 nasuwa wątpliwości typu:

co to znaczy „występuje gorączka”?

co to znaczy „szybka akcja serca”?

co to znaczy „silne dreszcze”?

co to znaczy „łatwe męczenie się”?

co to znaczy „nadmierna potliwość”?

Również globalne rozstrzygnięcia decyzyjne nie dają satysfakcji. Cóż bowiem znaczy „bardzo duże ryzyko” zapalenia wsierdza? czy też „średnie ryzyko”? i jak je mierzyć lub też porównywać?

Jeśli weźmiemy ponadto pod uwagę inne możliwe schorzenia kardiologiczne takie jak na przykład nadciśnienie, zawał serca, dusznica bolesna czy też arytmia (o podobnych symptomach), jak również fakt iż pacjent może cierpieć nie tylko na jedną chorobę z grupy kardiologicznych lecz na dwie lub więcej to wiarygodność takiej diagnozy staje się jeszcze bardziej problematyczna.

Jak też łatwo zauważyć we wszystkich rozpatrywanych węzłach decyzyjnych rozstrzygnięcie „trudno powiedzieć” nabiera szczególnego znaczenia. Rozstrzygnięcie tego typu, każdorazowo uruchomić musiałoby inny fragment ścieżki klinicznej np. żądając dodatkowego badania. Cząstkowe rozstrzygnięcie „trudno powiedzieć” przekładają się zatem w sposób bardzo istotny na globalne rozstrzygnięcie diagnostyczne. Przykład ten, choć bardzo uproszczony pokazuje z jednej strony wielką złożoność procesu diagnozowania a z drugiej strony konieczność obiektywizacji badań i pomiarów jak również traktowanie procesu diagnozowania w szerszym kontekście. Odpowiedzią nauki na te wyzwania są narzędzia jakie są stawiane lekarzom do dyspozycji głównie przez teorię zbiorów rozmytych, teorię zbiorów przybliżonych, metodologię sieci bayesowskich oraz teorię wzorców w analizie wielokryterialnej.

Pierwsze koncepcje zastosowania zbiorów rozmytych do modelowania zagadnień biologicznych i medycznych zostały

przedstawione w pracy Zadeha [20] już pod koniec lat sześćdziesiątych ubiegłego stulecia. Lata siedemdziesiąte i osiemdziesiąte to burzliwy rozwój zastosowań zbiorów rozmytych w diagnostyce medycznej Sanchez [14], [15], Wechsler [18], Smets [16]. Kolejny obszar zastosowań medycznych to analiza elektro-kardiogramów i elektroencefalogramów Albin [1] i wiele innych.

Zbiory rozmyte znalazły zastosowanie w diagnostyce medycznej głównie w modelowaniu zbiorów danych i symptomów medycznych (chorobowych).

Zbiorem rozmytym A w przestrzeni X jest zbiór par uporządkowanych:

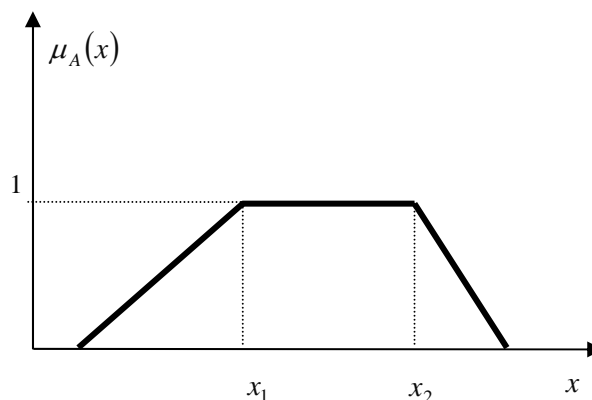
$$A = \{x, \mu_A(x) | x \in X\} \quad (2.1)$$

gdzie:

$\mu_A : X \rightarrow [0, 1]$ – funkcja przynależności elementów x do zbioru X

$\mu_A(x) \in [0, 1]$ – oznacza wielkość prawdopodobieństwa (stopień) przynależności elementu x do zbioru X .

W teorii zbiorów rozmytych [1], [21] najczęściej stosowane są następujące funkcje przynależności: trapezowa, trójkątna i tak zwana s - funkcja. Przykładową funkcję trapezową przedstawia kolejny rysunek:



Rys. 5. Funkcja trapezowa

Najważniejsze pojęcia używane w modelowaniu danych i symptomów medycznych to:

1) nośnik zbioru rozmytego

$$N(A) = \{x \in X | \mu_A(x) > 0\} \quad (2.2)$$

2) rdzeń zbioru rozmytego

$$R(A) = \{x \in X | \mu_A(x) = 1\} \quad (2.3)$$

3) wysokość zbioru rozmytego

$$h(A) = \sup_{x \in X} \mu_A(x) \quad (2.4)$$

zbiór rozmyty A jest znormalizowany, jeśli $h(A) = 1$.

4) Relacja równości zbiorów

$$A = B \Leftrightarrow \forall x \in X \mu_A(x) = \mu_B(x) \quad (2.5)$$

5) Relacja zawierania się zbiorów

$$A \subset B \Leftrightarrow \forall x \in X \mu_A(x) \leq \mu_B(x) \quad (2.6)$$

Omówione wyżej pojęcia dotyczące zbioru rozmytego wykorzystywane są do modelowania procesu rozpoznawania choroby. Uproszczony model choroby możemy wprowadzić następująco [3], [4]:

Symbolem $M = \{1, \dots, m, \dots, M\}$ oznaczmy zbiór wyróżnionych jednostek chorobowych. **Chorobę $m \in M$ (jednostkę chorobową)** oraz jej „nasilenie” można zdefiniować poprzez zakresy chorobowych wartości parametrów czyli wyników badań i symptomów [4],[5].

Ogólnie symbolem $N(m), m \in M$ możemy oznaczać zbiór parametrów i symptomów świadczących o chorobie $m \in M$.

Symbolem $C_n^m = [c_n^m, \overline{c_n^m}]$ oznaczamy zbiór wartości chorobowych parametru $n \in N$ w chorobie $m \in M$.

Tworząc w następnej kolejności iloczyn kartezyński zbiorów C_n^m otrzymamy model jednostki chorobowej $m \in M$ w postaci zbioru C^m . Tak więc:

$$C^m = C_1^m \times \dots \times C_n^m \times \dots \times C_{N(m)}^m \quad \text{gdzie } N(m)$$

łączna liczba parametrów (symptomów) świadczących o chorobie $m \in M$

$$C^m = \{c^m \in R^{N(m)} \mid c_n^m \in C_n^m, n \in N(m)\} \quad (2.7)$$

jest obrazem „chorobowych wartości” parametrów (symptomów) choroby $m \in M$.

Zachodzi przy tym [4],[5]:

$$C_n^m \subset S_n, m \in M, n \in N$$

W oparciu o zbiory C^m można utworzyć tzw. wzorce chorób $W(m) \subset C^m, m \in M$.

Idea zastosowania teorii zbiorów rozmytych w KSWDM polega na traktowaniu poszczególnych zbiorów C_n^m jako zbiorów rozmytych, a w konsekwencji również modelu C^m choroby $m \in M$ jako choroby rozmytej (zbioru rozmytego).

Jeśli stan zdrowia pacjenta na konkretnym etapie ścieżki klinicznej jest określony jako $x = (x_1, \dots, x_n, \dots, x_N)$ to proces diagnozowania polegać będzie na określeniu wartości funkcji przynależności poszczególnych jego aktualnych

parametrów zdrowotnych do odpowiednich zbiorów chorobowych wartości tych parametrów gdzie:

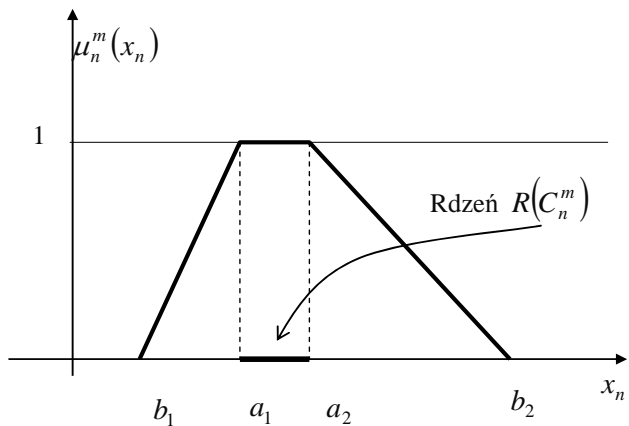
x_n – wartość parametrów n pacjenta x

μ_n^m – funkcja przynależności konkretnej wartości parametru n do zbioru wartości tego parametru chorobowego w chorobie m

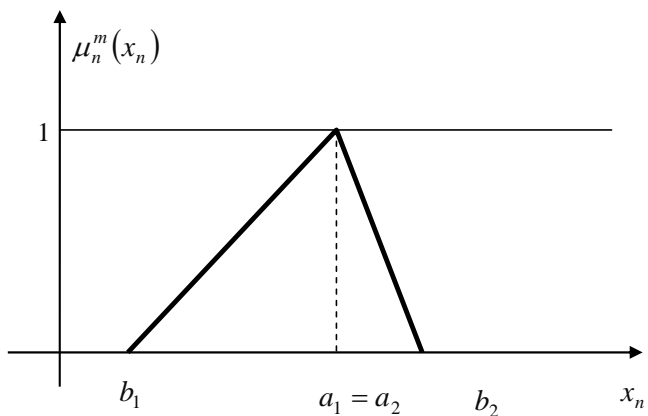
X_n – zbiór, w którym został zdefiniowany zbiór wartości chorobowych parametru $n \in N$.

W praktyce prawie zawsze $X_n \subset R^1, n \in N$

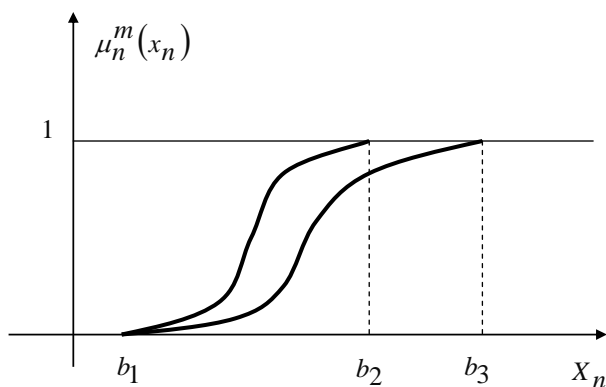
Przykłady najczęściej stosowanych funkcji przynależności przedstawiają poniższe rysunki:



Rys. 6. Trapezowa funkcja przynależności wartości parametru n w chorobie m



Rys. 7. Trójkątna funkcja przynależności wartości parametru n w chorobie m (rdzeń 1-elementowy)



Rys. 8. Funkcje przynależności typu S wartości parametru n w chorobie m

Nośnikiem zbioru rozmytego (parametru n w chorobie m) w przypadku funkcji trapezowej będzie zbiór:

$$R(C_n^m) = \{x_n \in X_n \mid \mu_n^m(x_n) > 0\} = \\ = \{x_n \in R^1 \mid b_1 < x_n < b_2\}$$

Rdzeniem (wzorcem parametru n w chorobie m) będzie natomiast zbiór

$$R(C_n^m) = \{x_n \in R^1 \mid \mu_n^m(x_n) = 1\} = \\ = \{x_n \in R^1 \mid a_1 \leq x_n \leq a_2\}$$

W ramach nośnika można definiować różne stopnie zagrożenia chorobą m określając odpowiednie obszary zagrożeń $Z(C_n^m)$, które będą podzbiorami nośnika $N(C_n^m)$.

Kompletna diagnoza choroby $m \in M$ będzie jednak możliwa do określenia po wyznaczeniu łącznej wartości funkcji przynależności $\mu^m(x)$ dla stanu zdrowia pacjenta opisanego wektorem $x = (x_1, \dots, x_n, \dots, x_N)$, czyli po określeniu wszystkich wartości $\mu^m(x), n \in N$. Będzie ona pewną funkcją $d(\mu_n^m(x), n \in N)$. I tu pojawia się pierwsza istotna trudność w stosowaniu tej koncepcji modelowania w zagadnieniach diagnostyki medycznej. Zgodnie z teorią zbiorów rozmytych [16], [20] funkcja ta ma następującą postać:

$$d(\mu_n^m(x), n \in N) = \mu^m(x) = \min_{n \in N} \mu_n^m(x_n)$$

co przeczy „zdroworozsądkowemu” podejściu i opiniom ekspertów. Zastosowanie bowiem powyższej konstrukcji funkcji d powoduje wyeliminowanie z możliwości wpływania na

ostateczną diagnozę wszystkich innych parametrów chorobowych mających większą wartość funkcji przynależności niż ta minimalna. Znane w literaturze inne próby modelowania funkcji d (w oparciu o sumę zbiorów rozmytych lub specjalnie definiowaną sumę ważoną) niosą ze sobą również zbyt duży „ładunek subiektywizmu” by można było je stosować w modelowaniu tak ważnego zagadnienia jakim jest diagnostyka medyczna. Funkcje trapezowa i funkcje typu S są najczęściej wykorzystywane w modelowaniu rozmytym procesu diagnozowania. Kluczowym problemem tej koncepcji modelowanie jest tzw. adekwatność funkcji przynależności [20]. Ich kształt ma kapitalne znaczenie w uzyskaniu poprawnej diagnozy. Pomiar empiryczny stopni przynależności jest niezwykle trudny i nacechowany przesłankami subiektywizmu. Zagadnienie to wchodzi w zakres tzw. skalowania psychologicznego. Brak adekwatnych funkcji przynależności a przede wszystkim brak skutecznych metod ich empirycznego wyznaczenia jest główną przyczyną, iż ta koncepcja modelowania węzłów decyzyjnych w ścieżkach klinicznych rzadko wychodziła poza modele poglądowe mające znaczenie głównie dydaktyczne.

3. Koncepcja modelowania systemu wspomagania decyzji medycznych oparta na teorii zbiorów przybliżonych

Teoria zbiorów przybliżonych powstała na początku lat osiemdziesiątych. Jej autorem jest Zdzisław Pawlak [11], [12]. Metodologia zbiorów przybliżonych zyskała bardzo dużą popularność, szczególnie w obszarze eksploracji danych, złożonych zadaniach klasyfikacji oraz w komputerowych systemach wspomagania decyzji. Z racji swojej specyfiki jest niezwykle obiecująca w obszarze Komputerowych Systemów Wspomagania Decyzji Medycznych [10], [11], [12].

Zbiory przybliżone są zbiorami definiowanymi w tzw. przestrzeni dyskretnej. Dyskretyzacja przestrzeni następuje poprzez określenie zbioru elementarnego, którego „wielkość” zależy od stopnia szczegółowości przybliżenia rozpatrywanych zbiorów. Elementy zawarte w zbiorze elementarnym są między sobą nierozróżnialne w określonym aspekcie. Elementy te posiadają wartości wszystkich cech takie same jak cały zbiór elementarny. Informacja repozytoryjna w koncepcji modelowania opartej na teorii zbiorów

przybliżonych przechowywana jest w postaci stabelaryzowanej w zakresie tak obiektów (pacjentów) jak i wartości cech (atrybutów) chorobowych. Jest to „informacja wytworzona” przez ekspertów. Systemy te operują na wartościach dyskretnych. Jeśli dane dotyczące cech (atrybutów) są ciągłe wówczas w fazie początkowej muszą zostać przekształcone do postaci dyskretnej.

Istota koncepcji zbiorów przybliżonych oparta jest na zdefiniowaniu tzw. klas równoważności obiektów (pacjentów) ze względu na identyczne wartości cech (symptomów) określonego podzbioru rozważanych cech chorobowych.

Jeśli X to skończony zbiór pacjentów, a N skończony zbiór symptomów chorobowych (cech atrybutów) to relację równoważności w sensie podzbioru $B \subset N$ cech chorobowych zapiszemy następująco:

$$IND(B) = \{(x, y) \in X \times X \mid n \in B \ f_n(x) = f_n(y)\} \quad (3.1)$$

gdzie:

$f_n(x)$ – wartość n -tego atrybutu (cechy) obiektu $x \in X$.

Klasą równoważności ze względu na podzbiór B cech chorobowych zapiszemy jako zbiór:

$$[x]_{IND(B)} = \{y \in X \mid (x, y) \in IND(B)\} \quad (3.2)$$

Jest to podzbiór tych elementów ze zbioru X które mają te same wartości cech chorobowych w grupie cech z podzbioru B . Są zatem nierozróżnialne. To pojęcie pozwala zdefiniować z kolei brzeg zbioru przybliżonego Y jako zbiór [11], [12]:

$$\delta_B(Y) = \overline{S_B}(Y) - \underline{S_B}(Y) \quad (3.3)$$

będący różnicą przybliżenia górnego i dolnego. Powstała w ten sposób możliwość zdefiniowania zbioru Y w oparciu o jego przybliżenie dolne i górne. Podobnie jak w przypadku teorii zbiorów rozmytych, teoria zbiorów przybliżonych zyskuje coraz to nowe możliwości w zakresie wnioskowania i podejmowania decyzji w sytuacji istnienia niepełnych lub częściowo sprzecznych danych.

Podjęcie to, głównie ze względu na stosunkowo niewielką potrzebę dysponowania subiektywnymi danymi typu „funkcje przynależności” oraz bazowaniu na zbiorach skończonych i skwantyfikowanych wartościach atrybutów (cech i symptomów chorobowych) rokuje znacznie większe nadzieje aplikacyjne w obszarze modelowania węzłów decyzyjnych w KSWDM i w konsekwencji modelowania

ścieżek klinicznych. Ujemną stroną tego podejścia jest konieczność szczegółowego opracowania danych służących budowie („wypełnieniu”) skomplikowanych tablic decyzyjnych przez ekspertów z dziedziny diagnostyki i leczenia poszczególnych chorób.

4. Koncepcja modelowania Systemu Wspomagania Decyzji Medycznych oparta na metodzie sieci bayesowskich

W wielu modelach systemów wspomagania decyzji medycznych zakłada się, że badany pacjent choruje tylko na jedną chorobę co skutkuje założeniem, że jednostki chorobowe objęte modelem wzajemnie się wykluczają. Założenie to bardzo upraszcza metody wnioskowania lecz niestety jest bardzo upraszczające i ograniczające. W odróżnieniu od dwóch poprzednich podejść do modelowania repozytorium, metodologia sieci bayesowskich pozwala uniknąć tego tak bardzo ograniczającego założenia [8], [9], [10].

Sieci bayesowskie [10], [8] są swoistego rodzaju probabilistycznymi modelami graficznymi. Najczęściej sieć bayesowska jest przedstawiana jako graf acykliczny skierowany. Głównym problemem przy konstruowaniu sieci bayesowskiej jest określenie jej struktury oraz określenie parametrów probabilistycznych w postaci prawdopodobieństw warunkowych. Idea modelowania opartego na sieci bayesowskiej bazuje na wyznaczeniu rozkładu prawdopodobieństwa a’posteriori wystąpienia poszczególnych chorób pod warunkiem stwierdzenia konkretnych wartości parametrów (symptomów) chorobowych $x \in X$

$$p(m/x), \ m \in M. \quad (4.1)$$

Jest to swego rodzaju „odległość” zestawu wyników $x = (x_1, \dots, x_n, \dots, x_N)$ od jednostki chorobowej $m \in M$. Adekwatność diagnozy podjętej w ten sposób bardzo istotnie zależy od stopnia dokładności empirycznego wyznaczenia takich rozkładów warunkowych. Zakładając ostrożnie, iż ilość jednostek chorobowych sięga kilku tysięcy, ilość parametrów (symptomów) chorobowych kilkuset, a zbiór wartości każdego z nich średnio kilku – kilkudziesięciu wartości (po dyskretyzacji) otrzymamy prawdziwą skalę problemu. Nie mniej, jednak praktyka ostatnich dwudziestu lat potwierdziła, iż w bardzo wielu przypadkach, modelowanie procesów diagnostycznych oparte na sieciach

bayesowskich przyniosło dobre rezultaty [8], [10].

5. Koncepcja modelowania Systemów Wspomagania Decyzji Medycznych oparta na metodzie wzorców deterministycznych

Tej metodzie modelowania poświęcono między innymi prace [2], [3], [4].

Ogólnie filozofia tego podejścia polega na porównaniu otrzymanych wyników badań medycznych pacjenta z wzorcami odpowiednich chorób. Szczególnym przykładem tego podejścia jest propozycja zawarta w [2], [3]. Polega ona na zdefiniowaniu stanu zdrowia pacjenta w tzw. przestrzeni życia oraz zdefiniowaniu w tej samej przestrzeni wzorców jednostek chorobowych.

Kolejnym krokiem jest zdefiniowanie odpowiedniej miary odległości stanu zdrowia pacjenta od wyróżnionego podzbioru chorób. Istotnym problemem w tym podejściu (sądzę, że również we wcześniej omawianych) jest problem obiektywnej i adekwatnej normalizacji zbiorów wartości poszczególnych cech i symptomów chorobowych.

Jest to jednak głównie problem matematyczno – techniczny. Problem wyboru „techniki mierzenia” odległości w przestrzeni życia jest w zasadzie rozwiązany [2], [19] i może być stosunkowo łatwo weryfikowany na podstawie danych a’posteriori.

6. Podsumowanie

Analizując pozytywne i negatywne aspekty przedstawionych w pracy koncepcji modelowania repozytorium medycznego oraz niezwykle bogatą literaturę z zakresu przykładów zastosowania poszczególnych koncepcji modelowania, należy stwierdzić iż podejście oparte na teorii zbiorów rozmytych doczekało się najmniej konkretnych aplikacji, szczególnie kompleksowych. Stąd też w dalszych pracach w ramach realizowanego projektu zostanie pominięte. Podejścia oparte na teorii zbiorów przybliżonych jak również bazujące na metodyce sieci bayesowskich wydają się bardziej obiecujące i są rekomendowane do dalszych badań.

Metoda bazująca na wskaźnikach odległościowych zdaniem autora opracowania jest najbliższa aplikacji. Rozwiązanie problemu normalizacji charakterystyk medycznych jak też wybór najbardziej adekwatnej metody mierzenia odległości w przestrzeni życia otworzyłyby drogę do jej szybkiej aplikacji.

7. Bibliografia

- [1] M. Albin, *Fuzzy sets and their applications to medical diagnosis*, Berkely, 1975.
- [2] A. Ameljańczyk, *Optymalizacja wielo-kryterialna w problemach sterowania i zarządzania*, Ossolineum, 1984.
- [3] A. Ameljańczyk, „Analiza specyfiki Komputerowych Systemów Wspomagania Decyzji Medycznych w kontekście modelowania i algorytmizacji procesów decyzyjnych”, I Krajowa Konferencja „Systemy Komputerowe i Teleinformatyczne w Służbie Zdrowia”, Warszawa, wrzesień, 2009.
- [4] A. Ameljańczyk, „Matematyczny model przestrzeni życia w komputerowym systemie wspomagania decyzji medycznych”, I Krajowa Konferencja „Systemy Komputerowe i Teleinformatyczne w Służbie Zdrowia”, Warszawa, wrzesień, 2009.
- [5] A. Ameljańczyk, „O pewnej koncepcji modelowania repozytorium medycznego”, WAT, Warszawa, kwiecień 2009, POIG.01.03.01-00-145/082009.
- [6] ANSI HL7, <http://www.hl7.org>.
- [7] J. Błaszczykowski, K. Krawiec, R. Słowiński, J. Stefanowski, Sz. Wilk, „Wspomaganie decyzji i komunikacji w systemach telemedycznych”, Politechnika Poznańska, Poznań, 2006.
- [8] J. Makal, „System ekspertowy do wspomagania diagnozy łagodnego przerostu prostaty”, *Pomiary Automatyka i Robotyka*, 7-8, 2004.
- [9] „Medyczne Systemy Ekspertowe”, <http://WWW.computer.privateweb.at/judith/links3.htm>
- [10] A. Oniśko i inni, „HEPAR I HEPAR II – komputerowe systemy wspomagania diagnozowania chorób wątroby”, XII Konferencja Biocybernetyki i Inżynierii Biomedycznej, Warszawa, listopad, 2001
- [11] Z. Pawlak, *Rough Sets*, *International Journal of Computer and Information Sciences*, vol. 11, 341-356, 1965.
- [12] Z. Pawlak, *Systemy informacyjne – podstawy teoretyczne*, WNT, Warszawa, 1983.
- [13] Resultmaker, *Workflow patterns of the Online Consultant v. 1.1.*, *Opracowanie wewnętrzne*, Kopenhaga, 2006.
- [14] E. Sanchez, „Inverses of fuzzy relations. Application to possibility distributions and medical diagnosis”, *Proc. IEEE Conf. Decision and Control*, USA 1977.

- [15] E. Sanchez, *Medical diagnosis and composite fuzzy relations. Advances in fuzzy sets theory and applications*, North-Holland, 1979.
- [16] P. Smets, „Medical diagnosis: Fuzzy sets and degrees of belief”, *Fuzzy Sets and Systems*, vol. 5, 1981.
- [17] „Ścieżki kliniczne jako dynamiczne środowisko dostępu do informacji medycznej pacjenta”, wersja 0.8., Zintegrowany System Informacji Medycznej o Pacjencie, Bielsko-Biała, Kraków, luty 2008.
- [18] H. Wechsler, „Applications of fuzzy logic to medical diagnosis”, *Proc. Symp. on Multiple – Valued Logic*, Logan 1975.
- [19] L.P. Yu, G. Leitmann, *Compromise solutions, domination structures and Salukwadze’s solution*, JOTA, vol. 13, 1974
- [20] L.A. Zadeh, „Fuzzy Sets”, *Information Control*, vol. 8, (338-353), 1965.

Matematyczne aspekty modelowania pajęczynowego obiektów

A. AMELJAŃCZYK

e-mail: aameljanczyk@wat.edu.pl

Instytut Systemów Informatycznych
Wydział Cybernetyki WAT
ul. S. Kaliskiego 2, 00-908 Warszawa

W pracy omówiono możliwości wykorzystania tzw. metod modelowania pajęczynowego do modelowania i analizy jakościowej obiektów złożonych w procesach eksploracji danych. Zdefiniowano takie pojęcia jak: N-wymiarowa przestrzeń pajęczynowa, model pajęczynowy obiektu, adekwatność modelu oraz niektóre charakterystyki eksploracyjne modelu.

Słowa kluczowe: eksploracja danych, matematyczny model obiektu, modelowanie pajęczynowe, model pajęczynowy obiektu, model M-dokładny, dokładność modelu, adekwatność modelu.

1. Wprowadzenie

Matematyczne modele obiektów (systemów) służą najczęściej analizom jakościowym i porównawczym. Mogą być też podstawą budowy modeli optymalizacyjnych jak również podstawą definiowania funkcji rankingowych [1], [3]. Formalne modele opisowe znalazły też duże zastosowania w procesach i metodach eksploracji danych [5], [6], [7]. Obszarem szczególnie interesującym z tego punktu widzenia jest modelowanie i analiza danych medycznych, modelowanie stanu zdrowia pacjenta i tzw. jednostek chorobowych, wykorzystywane w komputerowych systemach wspomagania decyzji medycznych [2], [3], [9].

W procesie modelowania matematycznego obiektów bardzo ważną rolę odgrywa cel modelowania [4], [5]. Z celu modelowania wynikają przede wszystkim wymogi dotyczące konkretnego, formalnego języka modelowania oraz wymagana dokładność (adekwatność) modelu. Dysponując oryginałem obiektu o nazwie $x \in X$, w zależności od celu modelowania, możemy zbudować wiele różnych modeli $y = F(x)$, tego samego obiektu.

Zbiór nazw modelowanych obiektów, na ogół jest utożsamiane z pewnym podzbiorem X zbioru liczb naturalnych \mathbb{N} . Zbiór X jest wtedy zbiorem numerów obiektów. W szczególnych przypadkach zbiór X nie musi być skończonym podzbiorem liczb naturalnych. Takim przykładem może być zbiór możliwych „zestaw danych medycznych” [2], [3], [9].

Modelowany obiekt $x \in X$ charakteryzuje się na ogół wieloma różnymi cechami cząstkowymi, które decydują o jego

globalnych (systemowych) własnościach. Szczegółowość (adekwatność) modelu – jakkolwiek definiowana jest oczywiście funkcją liczby cech (własności), które zostały uwzględnione w modelu. Im więcej cech uwzględnimy, tym na ogół model będzie bardziej dokładny (szczegółowy) i tym lepiej będzie „odzwierciedlał” oryginał. Taki model zapewne będzie jednak bardziej kosztowny, skomplikowany oraz bardziej niewygodny do prowadzenia badań i analizy. Problem ustalenia ile cech i które z nich należy wziąć pod uwagę jest oczywiście problemem wyboru odpowiedniego kompromisu między dokładnością modelu a jego złożonością i kosztem. Intuicyjnie można zgodzić się z opinią, że w każdym procesie modelowania istnieje pewna graniczna liczba cech do uwzględnienia, powyżej której, przy zadanym celu modelowania przyrost „jakości modelu” jest pomijalnie mały.

2. Uproszczony model opisowy obiektu $x \in X$

Oznaczmy symbolem x pewien obiekt ze zbioru X obiektów. Załóżmy dalej, że obiekt ten posiada maksymalnie N cech (własności), które mogą mieć znaczenie jeśli chodzi o adekwatność (jakość) modelu obiektu $x \in X$ z punktu widzenia przyjętego celu modelowania.

Załóżmy, że cechy świadczące o jakości obiektu są uporządkowane według ich ważności, zgodnie z porządkiem naturalnym.

Oczywiście wielkość liczby N zależy od „stopnia złożoności” obiektów ze zbioru X co zapiszemy $N = N(X)$. Jeśli przykładowo

elementami zbioru X będą „proste objekty” np. rodzaje bloków budowlanych, gatunki serów, itp. to maksymalna liczba wyróżnionych cech może sięgać kilku.

Jeśli będą to natomiast złożone urządzenia elektroniczne czy też złożone mechanizmy, to liczba cech sięgać może przykładowo kilkunastu.

W przypadku modelowania bardzo złożonych obiektów jak np. stanu zdrowia pacjenta [3], [6] liczba ta może sięgać kilkuset i więcej.

Teoretycznie możemy rozpatrywać procesy modelowania gdzie liczba $N \rightarrow \infty$.

Symbolem \aleph oznaczymy zbiór $\{1, \dots, n, \dots, N\}$ zaś symbolem \mathcal{M} – zbiór $\{1, \dots, m, \dots, M\} \subset \aleph$.

Określenie 2.1.

Modelem M -dokładnym ($M \leq N$) obiektu x nazwiemy model uwzględniający tylko M spośród N , najważniejszych cech obiektu $x \in X$.

Określenie 2.2.

Modelem dokładnym obiektu $x \in X$ nazwiemy model uwzględniający maksymalną liczbę N cech świadczących o jego jakości.

Określenie 2.3.

Funkcją modelowania opisowego nazywać będziemy funkcję $F : X \rightarrow \mathfrak{R}^N$, przyporządkowującą każdemu modelowanemu obiektowi $x \in X$ ciąg wartości poszczególnych jego cech. Tak więc modelem obiektu $x \in X$ będzie jego obraz

$$F(x) = (F_1(x), \dots, F_n(x), \dots, F_N(x)) \in \mathfrak{R}^N \quad (2.1)$$

dany funkcją modelowania F

gdzie $F_n(x)$ – wartość n -tej cechy modelowanego obiektu $x \in X$.

Założmy dalej, że dla każdego $n \in \aleph$

$$w_n \leq F_n(x) \leq w^n, \quad x \in X \quad (2.2)$$

Fakt ten będziemy zapisywać:

$$y_n = F_n(x) \in [w_n, w^n] \subset \mathfrak{R}^1, \quad n \in \aleph, x \in X \quad (2.3)$$

Ciągi liczbowe $y = (y_1, \dots, y_n, \dots, y_N) = F(x)$ nazywać też będziemy „danymi” o obiektach $x \in X$.

Niech przykładowo $N = 15$, $M = 9$. Modelem dokładnym obiektu $x \in X$ jest

$$F(x) = (1, 2, 1, 1, 3, 4, 8, 7, 2, 5, 4, 3, 1, 0, 0) \in \mathfrak{R}^{15}$$

zaś modelem M -dokładnym

$$F^M(x) = (1, 2, 1, 1, 3, 4, 8, 7, 2) \in \mathfrak{R}^9$$

Zadanie matematycznego modelowania opisowego obiektów x ze zbioru X możemy formalnie zapisać jako trójkę uporządkowaną:

$$Z_0 = (\mathfrak{R}^N, X, F) \quad (2.4)$$

Obraz zbioru Y zbioru X będzie zatem zbiorem modeli obiektów $x \in X$.

$$Y = F(X) = \{y = F(x) \in \mathfrak{R}^N \mid x \in X\} \subset \mathfrak{R}^N \quad (2.5)$$

Klasą równoważności $X(y)$ obiektów ze względu na ustalone wartości cech $y \in Y$ nazywamy przeciwobraz zbioru jednoelementowego $\{y\} \subset Y$

$$X(y) = F^{-1}(\{y\}) = \{x \in X \mid F(x) = y\} \quad (2.6)$$

Jeśli w zbiorze klas równoważności $\{X(y) \mid y \in Y\}$, istnieje przynajmniej jedna klasa o licznosci $|X(y)| > 1$, to zwiększenie ilości uwzględnianych cech może być przesłanką zwiększenia „dokładności modelowania”. Pojawia się tutaj interesujący problem określenia takiej liczby N (o ile istnieje), że dla każdego $y \in Y$, $|X(y)| = 1$

3. N-wymiarowa przestrzeń pajęczynowa

Założmy, że dane jest zadanie modelowania opisowego: $Z_0 = (\mathfrak{R}^N, X, F)$, którego wynikiem jest zbiór $Y = F(X)$ modeli opisowych obiektów $x \in X$.

Możliwość interpretacji graficznej (tzw. „zobrazowania”) takich modeli w celu analizy ich własności jest jednak bardzo ograniczona – maksymalnie do $N = 3$. O wiele większe możliwości graficznej interpretacji, a w konsekwencji analizy i „wydobycia” dodatkowej wiedzy z danych opisujących modelowane objekty daje tzw. modelowanie pajęczynowe (web modelling).

Określenie 3.1.

N -wymiarową, ograniczoną przestrzenią pajęczynową nazywać będziemy parę uporządkowaną

$$P_r = (S_r, N)$$

gdzie S_r jest pewnym zbiorem wyznaczanym na płaszczyźnie, np. w układzie współrzędnych biegunowych w następujący sposób:

$$S_r = \{(d, \varphi) \mid 0 \leq d \leq r, 0 \leq \varphi \leq 2\pi\} \quad (3.1)$$

Liczba $r > 0$ zwana jest promieniem (zakresem) przestrzeni, zaś para (d, φ) to współrzędne

punktu (elementu) zbioru S_r (d – odległość od tzw. bieguna, φ – kąt skierowany).

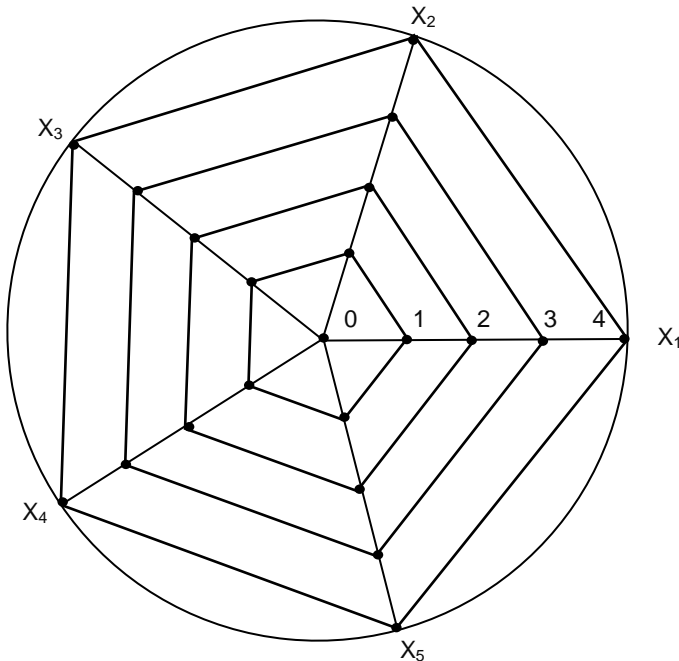
W przypadku gdy $r \rightarrow \infty$ otrzymamy N -wymiarową przestrzeń nieskończoną

$$P_\infty = (S_\infty, N).$$

Osie współrzędnych $0x_n, n=1, \dots, N$ przestrzeni pajęczynowej możemy wyznaczyć następująco:

$$0x_n = \left\{ (d, \beta_n) \mid 0 \leq d \leq r, \beta_n = \frac{2\Pi(n-1)}{N} \right\}, n \in \aleph \quad (3.2)$$

Na rys. 1. przedstawiona została pięciowymiarowa przestrzeń pajęczynowa z zaznaczoną „skalą odległości” (dla $r = 4$) oraz pięcioma „osiami współrzędnych” $0x_n, n \in \aleph = \{1, 2, 3, 4, 5\}$



Rys. 1. Pięciowymiarowa przestrzeń pajęczynowa o promieniu $r = 4$

Zbiór S_r bywa czasami definiowany wprost jako koło o promieniu r w układzie współrzędnych kartezjańskich:

$$S_r = \left\{ (s_1, s_2) \in \mathfrak{R}^2 \mid s_1^2 + s_2^2 \leq r^2 \right\} \quad (3.3)$$

Zamieniając współrzędne biegunowe na kartezjańskie otrzymamy:

$$S_r = \left\{ (s_1, s_2) \in \mathfrak{R}^2 \mid s_1 = d \cos \varphi, s_2 = d \sin \varphi, 0 \leq d \leq r, 0 \leq \varphi \leq 2\Pi \right\} \quad (3.4)$$

oraz osie współrzędnych:

$$0x_n = \left\{ (s_1, s_2) \in \mathfrak{R}^2 \mid s_1 = d \cos \varphi_n, s_2 = d \sin \varphi_n, 0 \leq d \leq r, \varphi_n = \frac{2\Pi(n-1)}{N} \right\}, n \in \aleph \quad (3.5)$$

Określenie 3.2.

Funkcją modelowania pajęczynowego F^P nazywać będziemy odwzorowanie typu:

$$F^P : Y \rightarrow 2^{S_r}$$

przyporządkowujące każdemu elementowi $y \in Y$ zbiór $S_N(y)$ w postaci wieloboku o wierzchołkach:

$$A_n(y) = (y_n, \beta_n), n \in \aleph \quad (3.6)$$

Symbolicznie fakt ten zapiszemy:

$$S_N(y) = [A_1(y), \dots, A_N(y)] \subset S_r \quad (3.7)$$

Wierzchołki wieloboku $S_N(y)$ leżą na poszczególnych osiach współrzędnych przestrzeni pajęczynowej (patrz tzw. wykresy gwiazdowe [6], [7]).

Określenie 3.3.

Modelem pajęczynowym obiektu y (a w konsekwencji obiektu x , takiego, że $F(x) = y$) nazywać będziemy zbiór

$$S_N(y) = F^P(y) \subset S_r.$$

Określenie 3.4.

Zadaniem modelowania pajęczynowego Z_P nazywać będziemy trójkę uporządkowaną

$$Z_P = (P_r, Y, F^P).$$

Wynikiem realizacji zadania modelowania pajęczynowego jest zbiór $F^P(Y) = Y_P$, którego elementami są modele pajęczynowe poszczególnych obiektów $y = F(x), x \in X$

$$Y_P = \left\{ S_N(y) \in 2^{S_r} \mid y \in Y \right\} \quad (3.8)$$

Przykład 1.

Niech $X = \{1, 2, 3\}, N = 5$.

Funkcja modelowania opisowego dana jest w tabeli 1. W tabeli tej zapisano również krańcowe wartości cech $w_n, w^n, n \in \aleph = \{1, 2, 3, 4, 5\}$

Wartość parametru r wyznaczamy następująco:

$$r = \max_{n \in \aleph} w^n = 6$$

Tabela 1.

$F(x)$ X	$F_1(x)$	$F_2(x)$	$F_3(x)$	$F_4(x)$	$F_5(x)$
1	2	3	$\frac{1}{2}$	1	4
2	1	1	2	4	1
3	3	4	6	4	5
w_n	1	1	$\frac{1}{2}$	1	1
w^n	3	4	6	4	5

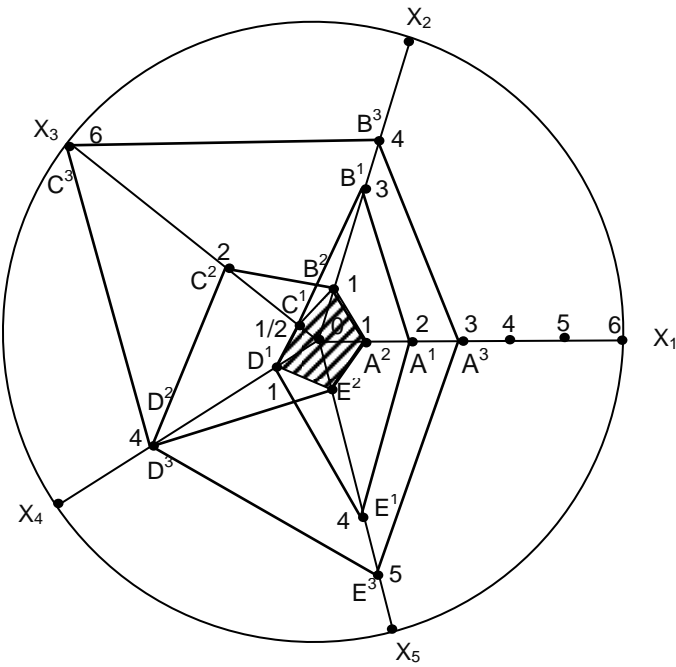
Pięciowymiarową przestrzeń z parametrem $r = 6$ zdefiniujemy jako parę uporządkowaną

$$P_6 = (S_6, 5),$$

gdzie

$$S_6 = \{(d, \varphi) | 0 \leq d \leq 6, 0 \leq \varphi \leq 2\Pi\}$$

Na rys. 2. przedstawiono modele pajęczynowe obiektów $x \in \{1, 2, 3\}$



Rys. 2. Modele pajęczynowe obiektów $x \in X$

Na rysunku tym zaznaczono również modele „sztucznych” obiektów [1]:

$$w_X = (w_1, \dots, w_5) = (1, 1, \frac{1}{2}, 1, 1)$$

oraz $w^X = (w^1, \dots, w^5) = (3, 4, 6, 4, 5),$

Zauważmy, że $S_5(w_X) \subset S_5(x)$ dla każdego $x \in X$ oraz, że $S_5(w^X) = S_5(3)$ a ponadto zachodzi $S_5(x) \subset S_5(w^X) = S_5(3)$ dla każdego $x \in X$.

Modelami pajęczynowymi obiektów $x \in \{1, 2, 3\}$ są odpowiednie zbiory $S_5(1), S_5(2), S_5(3)$.

Są to następujące wieloboki zaznaczone na rys. 2.:

$$S_5(1) = [A^1, B^1, C^1, D^1, E^1]$$

$$S_5(2) = [A^2, B^2, C^2, D^2, E^2]$$

$$S_5(3) = [A^3, B^3, C^3, D^3, E^3]$$

Modele „obiektów sztucznych” [1], [2] w_X

i w^X stanowią zbiory:

$$S_5(w_X) = [A^2, B^2, C^1, D^1, E^2] - \text{(zbiór ten został „zakreskowany”).}$$

$$S_5(w^X) = [A^3, B^3, C^3, D^3, E^3] = S_5(3)$$

Zbiory te stanowią odpowiednio kres dolny i kres górny zbioru $Y_P = F^P(X)$ w przestrzeni pajęczynowej z relacją inkluzji [1].

Zauważmy, że modele $S_5(1), S_5(2), S_5(3)$ różnią się między sobą „kształtem”, polem powierzchni, „wzajemnym usytuowaniem”, długością obwodu, liczbą boków itp. w przestrzeni $P_6 = (S_6, 5)$.

Takich dodatkowych charakterystyk modeli $S_N(x)$ może być wiele. Przykładowymi charakterystykami opisującymi dodatkowe właściwości $S_N(x)$ są następujące:

$p(S_N(x))$ – „pole powierzchni” zbioru $S_N(x)$

$g(S_N(x))$ – „środek ciężkości” zbioru $S_N(x)$

$b(S_N(x))$ – „długość obwodu” $S_N(x)$

$\omega(S_N(x))$ – „liczba boków” (wieloboku)

zbioru $S_N(x)$

$\delta_N(x)$ – „miara kąta wierzchołkowego”

trójkątów tworzących

zbiór $S_N(x)$ itp.

W przypadku ustalonych ciągów zbiorów $S_N(x), x \in X_k \subset X, k = 1, 2, \dots$ dodatkowymi charakterystykami mogą być „wzajemne usytuowania” elementów ciągów, wzajemne usytuowanie ich środków ciężkości, odległości środków ciężkości, części wspólne zbiorów $S_N(x), x \in X_k$, relacje inkluzji, różnica czy też suma zbiorów itp.

Wszystkie te charakterystyki przy odpowiedniej interpretacji mogą być bardzo cenne z punktu widzenia analizy jakościowej modelowanych obiektów jak też samego procesu modelowania.

Badanie tych charakterystyk pozwala bowiem „wydobyć” więcej informacji o obiektach $x \in X$ niż wynika to wprost z ich formalnych modeli opisowych $y = F(x), x \in X$ (danych w postaci ciągów liczb). Wprowadzając dodatkowo tzw. „gęstość cech” („ciężar

właściwy” cech) w postaci $\eta_n(x)$ możemy dokonywać analizy jeszcze bardziej szczegółowej.

W dalszej części pracy, z racji ograniczonej jej objętości zajmiemy się głównie charakterystyką modelu $S_N(x)$ w postaci $p(S_N(x))$ oraz charakterystyką kątową $\delta_N(x)$.

Pole powierzchni zbioru $S_N(x)$ jest pewną funkcją „danych opisowych” obiektu $x \in X$ ($F(x) = y$) oraz liczby N uwzględnionych cech.

Dla ustalonych N oraz $x \in X$ wartości charakterystyki $p(S_N(x))$ możemy określić następująco:

$$p(S_N(x)) = \alpha_N \left[\sum_{k=1}^{N-1} (y_k y_{k+1}) + y_N y_1 \right] \quad (3.9)$$

gdzie $\alpha_N = \frac{1}{2} \sin \frac{2\pi}{N}$ (3.10)

zaś $y_k = F_k(x), k = 1, 2, \dots, N$

Dodatkowe informacje dotyczące zbioru X modelowanych obiektów niosą też jak wcześniej wspomniano charakterystyki zbiorów:

$$S_N(w_X) \text{ i } S_N(w^X)$$

Odpowiednio pola powierzchni tych zbiorów określamy następująco:

$$p(S_N(w_X)) = \alpha_N \left[\sum_{k=1}^{N-1} (w_k w_{k+1}) + w_N w_1 \right] = \gamma_X(N) \quad (3.11)$$

$$p(S_N(w^X)) = \alpha_N \left[\sum_{k=1}^{N-1} (w^k w^{k+1}) + w^N w^1 \right] = \gamma^X(N) \quad (3.12)$$

Modele $S_N(w_X)$ oraz $S_N(w^X)$ odgrywają ważną rolę w analizie jakościowej obiektów $x \in X$, gdyż zbiory te posiadają następujące ogólne własności:

$$S_N(w_X) \subset S_N(x) \text{ dla każdego } x \in X \quad (3.13.)$$

$$S_N(x) \subset S_N(w^X) \text{ dla każdego } x \in X \quad (3.14)$$

a ponadto

$$\gamma_X(N) \leq p(S_N(x)) \leq \gamma^X(N), x \in X \quad (3.15)$$

Wartość $p(S_N(x))$ zależy oczywiście od kolejności uwzględnianych cech, stąd też w przypadku braku założenia ustalającego kolejność cech, stosowana jest charakterystyka $\bar{p}(S_N(x))$ będąca „uśrednieniem” pól powierzchni odpowiadających wszystkim

możliwym kombinacjom uporządkowań cech ($N!$).

4. Znormalizowane przestrzenie pajęczynowe

Niech $F(x) = (F_1(x), \dots, F_n(x), \dots, F_N(x))$ – funkcja modelowania taka, że $F : X \rightarrow Y \subset \mathfrak{R}^N$.

O funkcji tej założymy, że $F_n(x) \geq 0, n \in \aleph, x \in X$. Jeśli by tak nie było to należałoby od wartości funkcji $F_n(x)$ odjąć odpowiednią liczbę c_n dla każdego $x \in X$ (przesunięcie skali) [1], [4] w następujący sposób :

$$\bar{F}_n(x) = F_n(x) - c_n, x \in X,$$

gdzie $c_n = \min_{x \in X} \bar{F}_n(x), n \in \aleph$.

Dalej zakładając będziemy, że funkcja modelowania spełnia powyższy warunek.

Normalizacji funkcji modelowania można dokonać na wiele sposobów [1], [4].

Jednym z nich jest następujący sposób:

$$\overline{\overline{F}}_n(x) = \frac{\bar{F}_n(x)}{F_n} \quad (4.1)$$

gdzie

$$\bar{F}_n = \max_{x \in X} \bar{F}_n(x) \neq 0, n \in \aleph \quad (4.2)$$

W ten sposób znormalizowana funkcja modelowania ma taką własność, że

$$0 \leq \overline{\overline{F}}_n(x) \leq 1 \text{ dla każdego } n \in \aleph, x \in X \quad (4.3)$$

Zaś obiekty w_X i w^X mają postać:

$$w_X = (0, \dots, 0) \in \mathfrak{R}^N$$

$$w^X = (1, \dots, 1) \in \mathfrak{R}^N$$

Dalej będziemy zakładając, że funkcja F jest znormalizowana w sensie (4.3).

Określenie 4.1.

Znormalizowaną N -wymiarową przestrzenią pajęczynową nazywać będziemy parą uporządkowaną

$$P_r = (S_r, N)$$

gdzie $r = \max_{n \in \aleph} w^n = 1$

Dla uproszczenia zapisu, w przypadku przestrzeni znormalizowanych będziemy pisać $P = (S, N)$.

Przykład 2.

W tabeli 2. przedstawiono wartości znormalizowanej funkcji modelowania

z przykładu 1. (przesunięcie skali c jest następujące $c = \left(1, 1, \frac{1}{2}, 1, 1\right)$).

Tabela 2. Wartość znormalizowanej funkcji modelowania

$\begin{matrix} F(x) \\ X \end{matrix}$	$F_1(x)$	$F_2(x)$	$F_3(x)$	$F_4(x)$	$F_5(x)$
1	$\frac{1}{2}$	$\frac{2}{3}$	0	0	$\frac{3}{4}$
2	0	0	$\frac{6}{22}$	1	0
3	1	1	1	1	1
w_n	0	0	0	0	0
w^n	1	1	1	1	1

Na kolejnym rysunku przedstawione zostały znormalizowane modele pajęczynowe poszczególnych obiektów $x \in X$.

modelowanego obiektu lub też samego procesu modelowania.

Niniejsza praca jak już wcześniej wspomniano dotyczyć będzie charakterystyki $p(S_n(x))$, którą można głównie z racji jej konstrukcji (3.9) wykorzystać do badań jakościowych obiektów jak też wielokryterialnej analizy porównawczej oraz „charakterystyki katowej” będącej wprost, funkcją maksymalnej liczby uwzględnionych w modelu cech.

Założmy dalej, że dysponujemy modelem M -dokładnym obiektu $x \in X$. Maksymalna możliwa liczba cech, które można teoretycznie uwzględnić w przypadku modelowania obiektów ze zbioru X to $N(X)$. Oczywiście $M \leq N(X)$. Dla kompletności rozważań przyjmijmy, że $M = 0, 1, \dots, N(X)$. Jakość modelu $S_M(x)$ można określić definiując jego „rozbieżność” z oryginałem $x \in X$ lub zamiennie – „dokładność”.

Określenie 5.1.

Rozbieżnością modelu M -dokładnego $S_M(x)$ z oryginałem $x \in X$ nazywać będziemy liczbę $\overline{D}_M(x) = 2\Pi - M \delta_N(x)$, $M \leq N(X)$ gdzie M - liczba cech aktualnie uwzględnionych w modelu ,zaś $\delta_N(x) = \frac{2\Pi}{N(X)}$ graniczna miara katowa przyrostu „dokładności modelu” przy uwzględnieniu maksymalnej (dla elementów zbioru X) liczby cech.

Po przekształceniu mamy:

$$\overline{D}(x) = 2\Pi \left(1 - \frac{M}{N(X)}\right), \quad M \leq N(X) \quad (5.1)$$

$$0 \leq \overline{D}_M(x) \leq 2\Pi, \quad M \leq N(X) \quad (5.2)$$

Jeśli na $N(X)$ możliwych do uwzględnienia cech , uwzględnimy w modelu wszystkie, to rozbieżność takiego modelu z oryginałem będzie zerowa.

Jeśli uwzględnimy tylko jedną cechę to rozbieżność wyniesie

$$\overline{D}_1(x) = 2\Pi \frac{N(X)-1}{N(X)}.$$

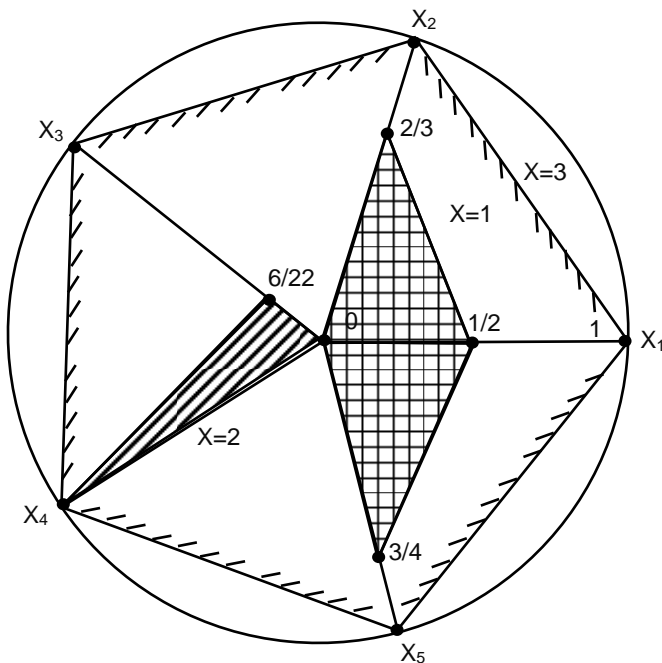
Określenie 5.2.

Stopniem (współczynnikiem) rozbieżności modelu nazywać będziemy liczbę

$$D_M(x) = \frac{\overline{D}_M(x)}{2\Pi} = 1 - \frac{M}{N(X)} \quad (5.3)$$

Mamy przy tym

$$0 \leq D_M(x) \leq 1, \quad M \leq N(X) \quad (5.4)$$



Rys. 3. Znormalizowane modele pajęczynowe obiektów $x \in X$

5. Problem szczegółowości i adekwatności modeli pajęczynowych

Wśród wielu dodatkowych charakterystyk modelu pajęczynowego $S_N(x)$ na szczególną uwagę zasługują charakterystyki, które mogą być wykorzystane do oceny jakościowej

Określenie 5.3.

Dokładnością modelu $S_M(x)$ nazywać będziemy liczbę

$$\overline{d}_M(x) = 2\Pi - \overline{D}_M(x) = 2\Pi \frac{M}{N(X)} \quad (5.5)$$

Stopień dokładności określimy następująco:

$$d_M(x) = \frac{\overline{d}_M(x)}{2\Pi} = \frac{M}{N(X)} \quad (5.6)$$

Zachodzi przy tym

$$0 \leq d_M(x) \leq 1, \quad M \leq N(X) \quad (5.7)$$

Gdy $M \rightarrow \infty$ zachodzi oczywiście $D_\infty(x) = 0$ i $d_\infty(x) = 1$

Zagadnienie adekwatności czy też rozbieżności modelu jest oczywiście problemem bardziej złożonym. Analizując charakterystykę $p(S_n(x))$ jak również „kształt” zbioru $S_N(x)$ można podjąć się prób definiowania innych, „bardziej związanych” z modelem obiektu $x \in X$ wskaźników charakteryzujących proces modelowania i sam model. Porównując kształty zbiorów $S_N(x)$ przy wzroście liczby N uwzględnianych cech, ich powierzchnię, obwód itp. można jakość modelu zdefiniować bardziej precyzyjnie. Kluczową rolę mogą odgrywać w tym podejściu graniczne wartości $S_\infty(x)$, $p(S_\infty(x))$ przy $N \rightarrow \infty$ jako swoistego rodzaju punkty odniesienia. Poniżej zostaną przedstawione pewne rozważania dotyczące możliwej interpretacji charakterystyki $p(S_\infty(x))$ w określaniu stopnia szczegółowości modelu $S_N(x)$ w zależności od wielkości liczby N . Rozważania te nie mają charakteru ogólnego i są słuszne jedynie dla obiektów $x \in X$ spełniających pewne szczególne założenia. Charakterystyki takie bowiem jak kształt zbioru $S_N(x)$, pole powierzchni, środek ciężkości itp. służą przede wszystkim analizie jakościowej obiektów $x \in X$ [1], [2], [3], a nie samego procesu modelowania.

Założmy, że mamy dane zadanie modelowania pajączynowego w przestrzeni znormalizowanej $Z = (P, X, F)$. Intuicyjnie, stopień szczegółowości modelu (opisu) obiektu $x \in X$ jak już było wspomniane powinien rosnać wraz z wielkością liczby M cech uwzględnionych w procesie modelowania ($M \leq N$), jakość (adekwatność) modelu powinna również posiadać podobną własność.

Wielkość pola powierzchni modelu $S_M(x)$, zgodnie z zależnością (3.9) i (3.10)

formalnie jest pewną funkcją liczby M uwzględnianych cech oraz wartości tych cech określonych jako ciąg liczb:

$$y = (F_1(x), \dots, F_m(x), \dots, F_M(x)) \in \mathfrak{R}^M$$

co zapiszemy

$$p(S_M(x)) = f(y, M) \quad (5.8)$$

Ustalając konkretną wartość M , możemy badać funkcję $f_M(y)$, $y \in Y$. Przy pewnych założeniach jakościowych odnośnie interpretacji $y_m = F_m(x)$, $m \in N$ wartości $f_M(y)$ może być interpretowana jako uogólniona jakość obiektu x , takiego że $F(x) = y$ [1], [6]. Można zatem ją wykorzystać do budowy tzw. funkcji rankingowej oraz innych funkcji decyzyjnych w tym w szczególności optymalizacji.

Z kolei ustalając $y = F(x)$, możemy badać zmienność $f_y(M)$ w zależności od $M = 1, 2, \dots, N$, co może pozwolić na ocenę zmian szczegółowości (dokładności) procesu modelowania, wynikających ze zmian liczby uwzględnionych cech. Przykładowo adekwatność modelu $S_M(x)$ możemy zdefiniować jako pewną funkcję $q(S_M(x))$ o postaci następującej:

$$q(S_M(x)) = \frac{p(S_M(x))}{p(S_N(x))}, \quad x \in X, \quad M = 1, \dots, N \quad (5.9)$$

Zauważmy, że dla $M = N$, $q(S_M(x)) = 1$.

6. Modelowanie obiektów w przestrzeni nieskończonej wymiarowej

W dalszej części opracowania zakładając będziemy, że dysponujemy znormalizowaną funkcją modelowania

$$F : X \rightarrow Y \subset \underbrace{[0, 1] \times \dots \times [0, 1]}_{M\text{-razy}} \quad (6.1)$$

Biorąc dowolny element $y \in Y$ możemy przedstawić jego model w przestrzeni pajączynowej P jako pewien podzbiór $S_M(y)$ zbioru S .

Głównymi charakterystykami tego modelu będą „szczegółowość” oraz „adekwatność” rozumiane następująco :

$$p(S_M(y)) \text{ oraz } q(S_M(y)) = \frac{p(S_M(y))}{p(S_N(y))} \quad (6.2)$$

Założmy, że chcemy zwiększyć „stopień szczegółowości modelu” oraz jego adekwatność

poprzez zwiększenie maksymalnej liczby N uwzględnianych cech.

Zwiększając liczbę N do nieskończoności ($N \rightarrow \infty$) otrzymamy nieskończenie wymiarową przestrzeń pajęczynową $P_\infty = (S, \infty)$.

Ustalając w konkretnym zadaniu modelowania liczbę $M \leq N$ najważniejszych cech otrzymamy model M-dokładny $S_M(y) \subset S$.

Określenie adekwatności $q(S_M(y))$ tak uzyskanego modelu $S_M(y)$ wymaga wyznaczenia granicznej wartości „stopnia szczegółowości” $p(S_N(y))$

$$\lim_{N \rightarrow \infty} p(S_N(y)) = p(S_\infty(y)) \quad (6.3)$$

o ile oczywiście granica taka istnieje.

O charakterystyce $p(S_N(y))$ możemy powiedzieć, że

$$0 \leq p(S_N(y)) \leq \Pi, \quad y \in Y, \quad N = 1, 2, \dots \quad (6.4)$$

Dla modelu $y = (1, \dots, 1)$, przy $N \rightarrow \infty$

$$p(S_N(y)) \rightarrow p(S_\infty(y)) = \Pi \quad (6.5)$$

$S_\infty(y)$ jest modelem obiektu $y \in Y$ w nieskończenie wymiarowej przestrzeni pajęczynowej (S_∞, ∞) .

Liczba $p(S_\infty(y))$ wyraża maksymalną (idealną) szczegółowość modelu, zaś adekwatność modelu jest równa 1.

Badając model przy ustalonej liczbie $M < \infty$ uwzględnianych cech, możemy określić też dodatkową jego charakterystykę:

$$\begin{aligned} d(S_M(y)) &= \frac{p(S_\infty(y)) - p(S_M(y))}{p(S_\infty(y))} = \\ &= 1 - \frac{p(S_M(y))}{p(S_\infty(y))} = 1 - q(S_M(y)) \end{aligned} \quad (6.6)$$

oczywiście przy $M \rightarrow \infty$, $d(S_M(y)) \rightarrow 0$.

7. Podsumowanie

Przedstawione powyżej rozważania stanowią próbę sformalizowania procesu modelowania obiektów, bazujące na tzw. przestrzeni pajęczynowej.

Graficzna interpretacja modeli pajęczynowych nawet dla dużych wartości liczby N jest możliwa i łatwa do implementacji komputerowej. Daje nowe przesłanki pogłębionej analizy jakościowej modelowanych obiektów. Jest typowym narzędziem analitycznym między innymi w procedurach eksploracji danych. Bardzo interesujące wyniki

można uzyskać wprowadzając w przestrzeni dodatkowe relacje np. inkluzji, określające wzajemne związki poszczególnych par modeli lub ich sekwencji. Wprowadzenie takiej relacji umożliwia porządkowanie obiektów wg różnych kryteriów lub wyznaczania klas równoważności, jak też konstruowanie funkcji rankingowych.

Dodatkowe charakterystyki w postaci wielkości pól powierzchni poszczególnych modeli graficznych oraz ich „wzajemnego usytuowania” i zależności w znormalizowanej przestrzeni pajęczynowej mogą być kolejnymi przesłankami ustaleń w zakresie wiedzy o analizowanych obiektach.

Pajęczynową przestrzeń modelową można potraktować też jako przestrzeń decyzyjną. Wprowadzenie do tej przestrzeni modelu preferencji decydenta może pozwolić na formułowanie, badanie i rozwiązywanie odpowiednio sformułowanych zadań optymalizacji.

8. Bibliografia

- [1] A. Ameljańczyk, *Optymalizacja wielokryterialna w problemach sterowania i zarządzania*, Ossolineum, Warszawa-Kraków, 1984.
- [2] A. Ameljańczyk, „Matematyczny model przestrzeni życia w komputerowym systemie wspomagania decyzji medycznych”, materiały I Krajowej Konferencji „Systemy komputerowe i teleinformatyczne w służbie zdrowia”, Warszawa, 2009.
- [3] A. Ameljańczyk, *O pewnej koncepcji modelowania repozytorium medycznego*, opracowanie wewnętrzne WAT w ramach projektu POIG.01.03.01-00-145/08/2009.
- [4] J. Gutenbaum, *Modelowanie matematyczne systemów*, PWN, 1987.
- [5] J. Kacprzyk, *Zbiory rozmyte w analizie systemowej*, PWN, 1986.
- [6] D. Hand, H. Mannila, P. Smyth, *Eksploracja danych*, WNT, Warszawa, 2005.
- [7] D.T. Larose, *Odkrywanie wiedzy z danych*, PWN, 2006.
- [8] Z. Pawlak, *Systemy informacyjne – podstawy teoretyczne*, WNT, Warszawa, 1983.
- [9] P. Smets, „Medical diagnosis: Fuzzy sets and degrees of belief”, *Fuzzy Sets and Systems*, vol. 5, 1981.
- [10] L.A. Zadeh, „Information Control”, *Fuzzy Sets*, vol. 8, 1965.

Analiza przydatności wybranych standardów do modelowania architektury systemu informatycznego dla służby zdrowia

T. GÓRSKI

e-mail: gorski@wat.edu.pl, tomasz.gorski@rightsolution.pl

Instytut Systemów Informatycznych
Wydział Cybernetyki WAT
ul. S. Kaliskiego 2, 00-908 Warszawa

Artykuł przedstawia wyniki analizy standardów BPMN, GELLO, UML, OCL, XML oraz HL7 w kontekście ich przydatności do modelowania architektury systemu informatycznego. Istotnym jest, że rozpatrywane standardy oceniane są pod kątem możliwości ich zastosowania do modelowania systemu informatycznego budowanego dla służby zdrowia. W artykule przedstawiono podstawowe cechy i własności każdego z rozpatrywanych standardów. Wskazano także własności każdego z rozpatrywanych standardów istotne z punktu widzenia architektury systemu informatycznego. W podsumowaniu zawarto rekomendację zestawu standardów do zastosowania przy modelowaniu architektury systemu informatycznego dla służby zdrowia.

Słowa kluczowe: architektura systemu informatycznego

1. Wprowadzenie

W artykule przedstawiono wyniki analizy standardów informatycznych w kontekście ich przydatności do modelowania architektury systemu informatycznego dla służby zdrowia. Analizie poddano następujące standardy:

- BPMN – Business Process Modelling Notation
- GELLO – Object-Oriented Query and Expression Language for Clinical Decision Support
- UML – Unified Modelling Language
- OCL – Object Constraint Language
- XML – Extensible Markup Language
- HL7 – Health Level Seven.

2. BPMN

BPMN jest standardem modelowania przebiegu procesów biznesowych, przepływów pomiędzy nimi oraz usług WWW. BPMN został opracowany przez organizację BPMI (ang. Business Process Management Institute). Jego głównym zadaniem jest dostarczenie sposobu opisu procesów i środowiska biznesowego zrozumiałego dla wszystkich użytkowników, od analityków biznesowych opracowujących początkowe zarysy procesów, aż po personel techniczny odpowiedzialny za wdrożenie technologii opracowanej na potrzeby tych procesów. Drugim, równie ważnym zadaniem, jest zapewnienie metody jednolitej wizualizacji języków wykonawczych procesów biznesowych

wykorzystujących XML, przez zastosowanie wspólnej notacji.

BPMN opisuje jeden diagram procesów biznesowych zwany BPD (ang. Business Process Diagram). Został on opracowany do realizacji dwóch celów. Po pierwsze jest łatwy w zrozumieniu i stosowaniu. Można go wykorzystać do szybkiego modelowania procesów biznesowych i jest zrozumiały dla użytkowników pozbawionych umiejętności technicznych. Po drugie dostarcza możliwości modelowania skomplikowanych i złożonych procesów biznesowych oraz może być przełożony na język wykonawczy procesów biznesowych.

W procesie modelowania przebiegu procesu biznesowego wystarczy zamodelować zdarzenia powodujące rozpoczęcie procesu, następnie działania, które muszą zostać podjęte do realizacji procesu, a na końcu możliwe rezultaty przebiegu procesu. Decyzje oraz rozgałęzienia procesów modeluje się za pomocą węzłów decyzyjnych. Na rysunku rys.1. znajduje się przykład diagramu BPD.

Ponadto, proces może zawierać w sobie procesy zstępujące, które z kolei mogą być reprezentowane w innym diagramie BPD podłączonym za pomocą linku do dowolnego symbolu procesu macierzystego. Jeśli proces nie zawiera żadnych procesów zstępujących to jest uznawany za proces najniższego poziomu i nazywany zadaniem.

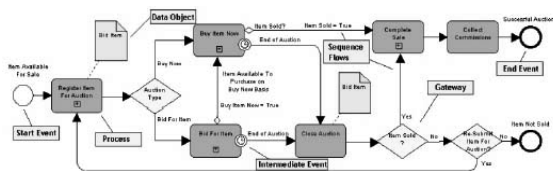


Figure 1. Simple BPMN Business Process Diagram for an on-line auction system.

Rys. 1. Diagram BPD dla systemu aukcji on-line

Zagłębiając się dalej w analizę biznesową można umieszczać zdarzenia i procesy w cieniowanych obszarach zwanych pulami, określających kto realizuje proces (rys. 2.). Pula może być z kolei podzielona dalej na tory. Pula zazwyczaj reprezentuje organizację, podczas gdy tor, jej określony dział lub pion.

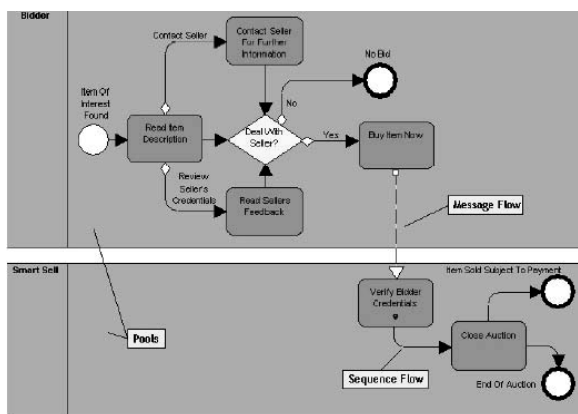


Figure 2. BPMN Business Process Diagram with processes drawn in pools.

Rys. 2. Diagram BPD z procesami pogrupowanymi w pulę

W trakcie modelowania procesów biznesowych niezbędne jest również modelowanie zdarzeń występujących w tych procesach i przedstawienie sposobu, w jaki zdarzenia wpływają na przebieg procesów. Zdarzenie powoduje uruchomienie procesu, występuje w trakcie przebiegu procesu lub kończy proces. BPMN dostarcza wyraźnie różnicującej notacji dla każdego ze wspomnianych typów zdarzeń.

Przy modelowaniu bardziej złożonych przebiegów procesów, takich jak usługi WWW B2B pojawia się potrzeba modelowania bardziej skomplikowanych zdarzeń, takich jak wiadomości, zależności czasowe (ang. timers), zasady biznesowe (ang. business rules) i błędy. BPMN umożliwia określenie typu aktywatora zdarzenia. Określenie typu aktywatora zdarzenia nakłada określone ograniczenia na modelowany przebieg procesu. Przykładowo zależność czasowa nie może kończyć przebiegu procesu, a przepływ wiadomości może mieć miejsce wyłącznie pomiędzy zdarzeniami wiadomości. Narzędzie modelowania wspierające model BPMN powinno automatycznie wymusić te zależności modelowania, które są specjalnymi przypadkami reguł biznesowych.

Zdarzenia mogą występować w trakcie realizacji procesu, powodując przerwanie procesu i powodując uruchomienie nowego procesu. Ewentualnie proces kończy się powodując zdarzenie i pośrednio uruchomienie nowego procesu. Można modelować te zdarzenia pośrednie (ang. intermediate events) przez umieszczenie symbolu zdarzenia bezpośrednio na procesie, z którym jest ono związane.

Istnieją trzy typy procesów – proces, proces zstępujący i zadanie.

Ponieważ diagram BPD w specyfikacji BPMN jest głównie zaprojektowany tak, aby był łatwy w odbiorze, więc aby ułatwić odbiorcom zrozumienie złożoności procesów, można graficznie ją zaprezentować przez umieszczenie miniaturki potomnych diagramów BPD w symbolach procesów. Dzięki temu można, oglądając diagram BPD określić, które procesy są złożone i dzielą się na procesy zstępujące niższych poziomów.

Uwidocznienie kolejności realizacji procesów osiąga się poprzez połączenie ich przepływami sekwencyjnymi (ang. sequence flow). Przepływ sekwencyjny jest stosowany do określenia kolejności działań w organizacji lub dziale. Zatem, jeśli diagram jest pogrupowany w pule i tory, to przepływy sekwencyjne łączą zdarzenia, procesy i punkty decyzyjne w obrębie torów i pul. BPMN udostępnia drugi rodzaj linii przepływu do modelowania, tzn. przepływ wiadomości (ang. message flow), który pozwala określić kolejność działań pomiędzy organizacjami lub pulami.

Jednym z celów opracowania specyfikacji BPMN jest umożliwienie modelowania przepływu informacji w relacjach B2B. W tym celu, diagram BPD oferuje metodę modelowania przepływów wiadomości. Tradycyjne diagramy przepływów procesów biznesowych pozwalają na modelowanie sekwencyjnych przepływów procesów od zdarzeń rozpoczynających proces do rezultatów końcowych. Diagram BPD wspiera przepływy sekwencyjne za pomocą przepływów wiadomości, co pozwala zobrazować wysyłanie i otrzymywanie wiadomości pomiędzy ludźmi czy organizacjami. Stanowi to istotny element obrazowania i rozumienia relacji między przedsiębiorstwami (B2B) i między przedsiębiorstwem a klientem (B2C).

Specyfikacja ta została zaprojektowana tak, aby umożliwić łatwe modelowanie typowych procesów biznesowych, a jednocześnie pozwolić na modelowanie procesów o dowolnie dużym stopniu złożoności.

3. GELLO

GELLO jest zorientowanym obiektowo językiem, który jest zbudowany na istniejących standardach. GELLO obejmuje zarówno zapytania jak i języki wyrażeń. GELLO wykorzystuje język Object Constraint Language (OCL), opracowany przez Object Management Group. Istotne składniki OCL zostały wybrane i zintegrowane w zapytaniach i językach wyrażeń GELLO, aby dostarczyć odpowiednich podstaw do manipulacji danymi medycznymi w celu podejmowania decyzji w ochronie zdrowia. Język GELLO może być używany do:

- budowania zapytań do wydobywania i manipulowania danymi z historii choroby
- konstruowania kryteriów decyzyjnych przez tworzenie wyrażeń dla szczególnych danych cecha/wartość – te kryteria mogą być używane w bazach wiedzy systemów wspomagania decyzji takich jak te zaprojektowane by dostarczyć ostrzeżeń i przypomnień, wskazówek, albo inna zasad decyzyjnych
- tworzenia wyrażeń, formuł, i zapytań dla innych aplikacji.

Zapytania i języki wyrażeń współdzielą model danych obiektowych dopóki wyrażenia przetwarzają dane dostarczone przez zapytania. Język zapytań został zaprojektowany w kontekście modelu zaproponowanego w HL7 CDSTC. Model ten proponuje wykorzystanie vMR, który dostarcza standardowego interfejsu do różnych systemów historii choroby. Język wyrażeń może być używany dla określania kryteriów decyzji albo abstrakcyjnych lub dostarczonych zbiorczych wartości. Obiektowe podejście pozwala na uzyskanie cechy rozszerzalności tego języka.

W celu ułatwienia procesu kodowania i oceny wyrażeń a co ważniejsze, maksymalizowania możliwości współdzielenia zapytań i wyrażeń, GELLO obejmuje podstawowe wbudowane typy danych. Ponadto dostarcza koniecznych mechanizmów dostępu do zasadniczego modelu danych.

Główną przeszkodą do dzielenia się wiedzą medyczną jest brak wspólnego formatu dla kodowania i manipulacji danymi. Pomimo, że składnia Arden Syntax zajęła się tym problemem przez odizolowywanie odniesień do lokalnych danych w nawiasach klamrowych w MLMs, to wciąż nie dostarcza mechanizmów dostępu do danych w sposób niezależny od formatu.

Wirtualna historia choroby (vMR), to obiektowe podejście zgodne z HL7 RIM, używające typowego modelu danych, jako

pośrednika do różnych systemów historii choroby. Pojęcie vMR zostało zaproponowane, jako podstawowy model dla przetwarzania danych pacjentów w kontekście systemów wspomagania decyzji. vMR jest udoskonaleniem modelu: Reference Information Model (RIM).

Podejście to dostarcza pierwszego podejścia w kierunku standardu wymiany, zarządzania i integracji danych medycznych. Jednakże, nadejście obiektowej wirtualnej karty choroby vMR jest niekompatybilne ze składnią Arden Syntax, ponieważ może ona przetwarzać tylko atomowe typy danych.

Faktem jest, że istnieje potrzeba języka do formułowania zapytań i wyrażeń w celu wydobywania i manipulowania danymi medycznymi. Język taki powinien być:

- niezależny od dostawcy
- niezależny od platformy
- zorientowany obiektowo i kompatybilny z vMR
- łatwy w czytaniu i używaniu
- rozszerzalny.

Najważniejsze cele i własności zaproponowanego języka GELLO, jako niezależnego od platformy standardu zapytań i wyrażeń, dla współdzielenia i manipulowania wiedzą w medycznym kontekście, to:

- GELLO jest skierowany dla medycznych aplikacji, które muszą korzystać z zapytań i języków wyrażeń dla wspomagania decyzji
- GELLO jest niezależny od dostawcy przez wykorzystywanie specyfikacji języka, który nie jest zależny od dostawcy
- GELLO jest niezależny od platformy
- GELLO dostarcza mechanizmów dostępu do danych przez obiektowy model danych, za pomocą ściśle typowanych wyrażeń, przez ogólnego przeznaczenia języki zapytań i wyrażeń
- GELLO jest językiem deklaratywnym
- GELLO jest rozszerzalny przez dodawanie nowych klas definiowanych przez użytkownika do podstawowego obiektowego modelu danych
- wszystkie metody manipulowania danymi muszą być wyraźnie określone w obiektowym modelu danych
- przez używanie określonego obiektowego modelu danych, takiego jak vMR, każda zasada decyzji albo wskazówka nie potrzebuje oddzielnego mechanizmu translacji elementów danych do/z środowisk dostarczających dane

- podejście zorientowane obiektowo umożliwia modularność, hermetyzację i rozszerzalność.

4. UML

Unified Modeling Language jest językiem do wizualizacji, specyfikacji, dokumentowania i budowania systemu informatycznego. Dzięki zastosowaniu UML można w sposób graficzny przedstawić różne spojrzenia na system: strukturalne, dynamiczne. Tworząc modele UML dokonujemy specyfikacji wymagań na system, szczególnie stosując model przypadków użycia i diagramy aktywności. Modelując system za pomocą diagramów UML jednocześnie dokumentujemy decyzje projektowe w jednoznaczny i zrozumiały dla całego zespołu sposób.

W ramach UML 2.0 występują następujące typy diagramów zachowania (przypadków użycia, aktywności, sekwencji, przeglądu interakcji, komunikacji, stanów, czasowy) oraz statyczne (klas, obiektów, pakietów, komponentów, struktury połączeń, wdrożeniowy).

Wersja UML 2.0 została opisana w precyzyjny i zwięzły sposób w [1].

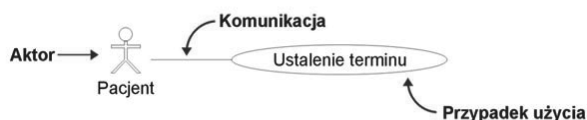
W kontekście architektury systemu informatycznego największe zastosowanie mają diagramy:

- Przypadków użycia
- Klas
- Sekwencji
- Komunikacji
- Aktywności
- Wdrożeniowy.

Diagramy przypadków użycia opisują, co robi system z punktu widzenia zewnętrznego obserwatora. Diagramy przypadków użycia pozostają w bliskim związku ze scenariuszami. Scenariusz to przykład tego, co się dzieje, kiedy ktoś wchodzi w interakcję z systemem. Oto scenariusz dla kliniki medycznej:

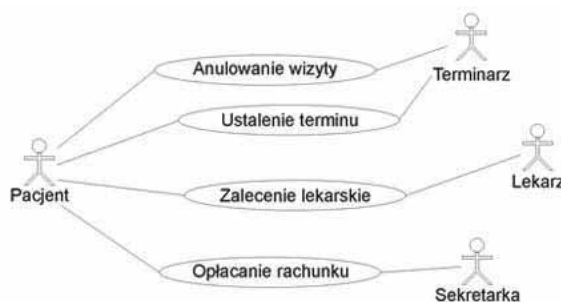
"Pacjent dzwoni do kliniki, aby umówić się na coroczne badania. Recepcjonistka znajduje najbliższy wolny termin w książce przyjęć i wyznacza badanie na tę datę".

Przypadek użycia to podsumowanie scenariuszy pojedynczego zadania lub celu. Aktor to ktoś albo coś, co inicjuje zdarzenia związane z tym zadaniem. Aktor po prostu określa rolę, którą odgrywa człowiek lub obiekt. Rys. 3. przedstawia przypadek użycia Ustalenie terminu badania w klinice medycznej. Aktor to Pacjent. Połączenie między aktorem a przypadkiem użycia to komunikacja.



Rys. 3. Przykład komunikacji między aktorem i przypadkiem użycia

Diagram przypadków użycia to zbiór aktorów, przypadków użycia i ich komunikacji (rys. 4.).



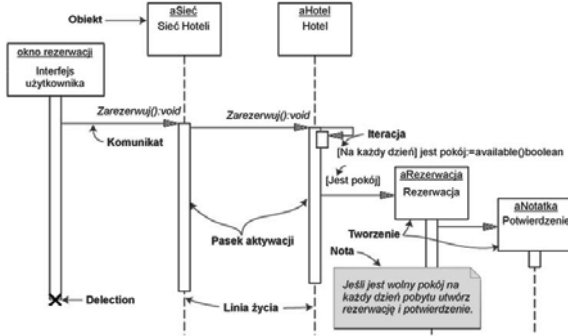
Rys. 4. Przykład diagramu przypadków użycia

Diagramy przypadków użycia mają trzy zastosowania:

- określanie funkcji (wymagań) - nowe przypadki użycia często generują nowe wymagania, kiedy system jest analizowany i projekt przybiera coraz wyraźniejszy kształt,
- komunikacja z klientami - prostota notacji sprawia, że diagramy przypadków użycia są dobrym sposobem porozumiewania się programistów z klientami,
- generowanie przypadków testowych - zbiór scenariuszy danego przypadku użycia może zasugerować sposoby testowania tych scenariuszy.

Diagram klas przedstawia ogólną panoramę systemu, pokazując klasy i ich wzajemne relacje. Diagramy klas są statyczne – pokazują elementy wchodzące w interakcje a nie działania zachodzące podczas interakcji.

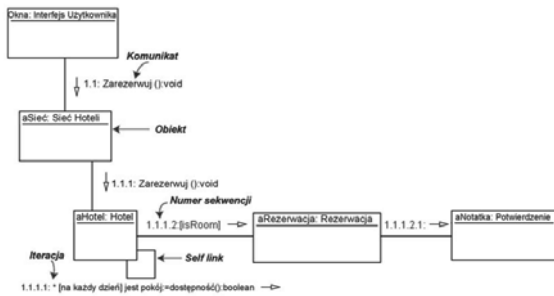
Diagram sekwencji to diagram interakcji, który szczegółowo pokazuje kolejność przekazywania komunikatów, a co za tym idzie, wywoływania operacji w obiektach. Na rys. 5. przedstawiono diagram sekwencji ilustrujący rezerwację hotelu. Obiektem inicjującym sekwencję komunikatów jest Okno rezerwacji.



Rys. 5. Przykład diagramu sekwencji

Każda przerywana pionowa linia to linia życia reprezentująca czas, przez który istnieje obiekt. Każda strzałka to przesłanie komunikatu. Strzałka zaczyna się od nadawcy, a kończy na pasku aktywacji komunikatu na linii życia odbiorcy. Na powyższym diagramie Hotel wykonuje autowywołanie, aby sprawdzić, czy dysponuje wolnym pokojem. Jeśli tak, to Hotel tworzy Rezerwację i Potwierdzenie. Wyrażenie w nawiasie kwadratowym, [], to warunek.

Diagramy komunikacji to również diagramy interakcji. Dostarczają tych samych informacji co diagramy sekwencji, ale skupiają się na rolach obiektów, a nie na czasach przesyłania komunikatów. Przykład diagramu komunikacji znajduje się na rys. 6. Na diagramie sekwencji role obiektów są wierzchołkami, a komunikaty – liniami łączącymi wierzchołki.



Rys. 6. Przykład diagramu komunikacji

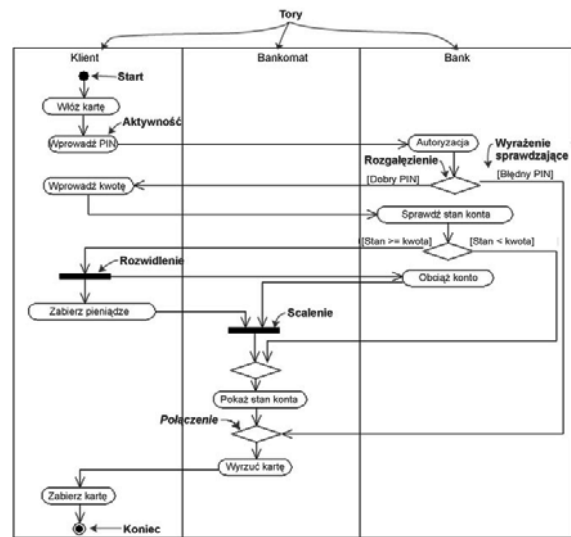
Prostokąty opisujące rolę obiektu są oznaczone nazwami klas lub obiektów (albo oboma nazwami). Nazwy klas są poprzedzone dwukropkiem (:).

Każdy komunikat na diagramie komunikacji ma numer sekwencji. Komunikaty najwyższego poziomu mają numer 1. Komunikaty na tym samym poziomie (wysyłane podczas tego samego wywołania) mają ten sam przedrostek dziesiętny oraz przyrostki 1, 2 itd. w zależności od tego, kiedy występują.

Diagram aktywności to inaczej diagram przepływu. Diagram aktywności skupia się na obiekcie przechodzącym pewien proces (albo na

procesie traktowanym jak obiekt). Diagram aktywności pokazuje wzajemne zależności między tymi operacjami.

Na rys. 7. przedstawiono przykład diagramu aktywności dla procesu "Pobieranie pieniędzy z konta bankowego za pomocą bankomatu".



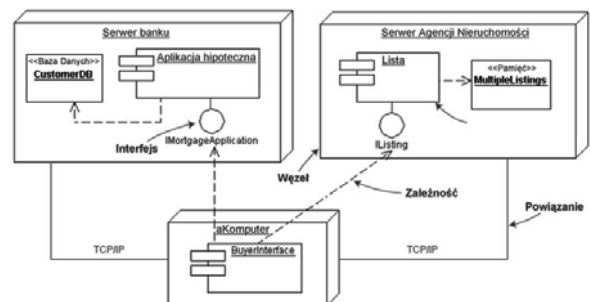
Rys. 7. Przykład diagramu aktywności

Diagramy aktywności można dzielić na tory obiektów. Tory określają, który obiekt jest odpowiedzialny za daną aktywność. Od każdej aktywności odchodzi przejście, łącząc ją z kolejną aktywnością.

Przejście może rozgałęziać się na dwa lub więcej wzajemnie wykluczających się przejść. Wyrażenia sprawdzające (w nawiasach kwadratowych []) opisują przejścia wychodzące z rozgałęzienia. Rozgałęzienie i jego późniejsze scalenie, które wskazują koniec rozgałęzienia, pojawiają się na diagramie jako puste romby. Przejście może rozwidlać się na dwie lub więcej równoległych aktywności. Rozwidlenie i późniejsze połączenie wątków wychodzących z rozwidlenia jest przedstawione, na diagramie, jako grube poziome linie.

Diagramy wdrożeń pokazują fizyczną konfigurację oprogramowania i sprzętu.

Diagram wdrożenia (rys. 8.) pokazuje związek między komponentami programowymi i sprzętowymi związanymi ze sprzedażą nieruchomości.



Rys. 8. Przykład diagramu wdrożeniowego

Fizyczny sprzęt składa się z węzłów. Każdy komponent należy do węzła. Komponenty są przedstawione, jako prostokąty z dwoma zakładkami w lewym górnym rogu.

5. OCL

OCL jest językiem wyrażeń, umożliwiającym formułowanie ograniczeń dla modeli obiektowych i innych artefaktów powstałych w czasie modelowania obiektowego. Ograniczenie to restrykcja nałożona na jedną lub więcej wartości (części) modelu lub systemu obiektowego.

Bertrand Meyer, jeden z pionierów stosowania ograniczeń, nazywa ograniczenia asercjami (ang. assertion) i definiuje, jako „wyrażenie znaczenia elementu”. Asercje w rozumieniu Meyera występują w trzech odmianach: warunki początkowe, warunki końcowe i niezmienniki.

Ian Graham używa zarówno terminu asercja, jak i reguła. Asercje to w jego rozumieniu „postać niezmiennika, warunku początkowego i warunku końcowego, który musi być spełniony, odpowiednio, gdy metoda jest wykonywana, wywoływana i zakończona”. Natomiast pojęcie reguły określa sposób wyrażania wzajemnego oddziaływania obiektów”.

James Rumbaugh definiuje ograniczenie, jako „zależność funkcjonalną między bytami modelu obiektowego”. W opracowanej przez niego metodzie analizy i projektowania (OMT) ograniczenia zawężają zbiór możliwych wartości bytów modelu obiektowego.

Grady Booch także używa pojęcia ograniczenia, rozumiejąc je, jako „wyrażenie opisujące warunek znaczeniowy, który musi być spełniony”. Podkreśla, że ograniczenie może być spełnione tylko wtedy, kiedy system jest w stanie stabilnym. Może dojść do przejściowych sytuacji, kiedy ograniczenia nałożone na system nie będą obowiązywały.

Język OCL pomaga wyrazić wspólny element wszystkich definicji i ustalić zrozumiałe i łatwe w użyciu standard umożliwiające projektantowi specyfikowanie tego co niezbędne.

Za pomocą warunków początkowych i końcowych można skutecznie określać operacje i metody. Używając wyrażeń OCL wewnątrz modelu UML, możemy wyspecyfikować warunki początkowe i końcowe operacji oraz metod dla wszystkich klas, typów i interfejsów. Zasada kryjąca się za takim sposobem postępowania nosi nazwę „projektowania według umowy” (ang. design by

contract). Może być stosowana w dowolnej metodzie obiektowej. Jest ponadto jednym z podstawowych paradygmatów inżynierii oprogramowania.

W terminologii obiektowej umowa jest sposobem jasnego i jednoznacznego przydzielania obiektowi odpowiedzialności (ang. responsibilities). Obiekt jest zobowiązany do wykonania usługi wtedy i tylko wtedy, kiedy są spełnione warunki. Umowa jest dokładną specyfikacją interfejsu obiektu, nazywanego dostawcą (ang. supplier). Natomiast obiekty, korzystające z oferowanych przez dostawcę usług, są nazywane klientami (ang. clients) lub konsumentami (ang. consumers).

Na interfejs obiektu składa się pewna liczba wykonywalnych operacji. Prawa obiektu-dostawcy są określane w postaci warunków początkowych, które muszą być spełnione tuż przed wywołaniem operacji. Obowiązki specyfikują warunki końcowe, które muszą być prawdziwe zaraz po zakończeniu operacji. Niedotrzymanie warunków początkowego lub końcowego oznacza naruszenie umowy. Warunki początkowe i końcowe są uzupełniane trzecim rodzajem ograniczenia: niezmiennikiem – restrykcją ściśle związaną z klasami, typami i interfejsami.

Niezmiennik jest ograniczeniem, które musi być zawsze spełnione przez wszystkie egzemplarze klas, typów lub interfejsów. Warunki początkowe i końcowe muszą być prawdziwe tylko w określonej chwili – odpowiednio przed i po wykonaniu operacji.

W języku UML powyższe trzy postacie ograniczeń zdefiniowano, jako standardowe stereotypy: <<invariant>>, <<precondition>>, <<postcondition>>.

Ograniczenia są jednoznaczne i zwiększają dokładność modelu lub systemu, do którego się odnoszą.

Gdy stosujemy język OCL do wyrażania ograniczeń, możemy używać analizatora składni OCL, aby upewnić się, co do poprawności sformułowania ograniczeń nałożonych na model. Taka weryfikacja pomaga opracować spójny i precyzyjny model lub system.

Ograniczenia wyrażane w języku deklaratywnym nie mają wpływu na stan systemu. Innymi słowy, stan systemu nie ulega zmianie z powodu takiej a nie innej wartości wyrażenia.

Dzięki kontroli typów wyrażenia OCL mogą być wartościowane podczas modelowania, przed uruchomieniem systemu.

Ograniczenia OCL służą podczas modelowania i specyfikowania, lecz nie są

opisem implementacji. Określają tylko, jakie twierdzenia są prawdziwe w wyidealizowanym, bezbłędnym systemie. Język OCL nie umożliwia wyrażania działań, które powinny zostać podjęte w wypadku naruszenia ograniczenia w niepoprawnej realizacji systemu.

6. XML

XML jest językiem utworzonym przez World Wide Web Consortium (W3C). W XML można tworzyć własne znaczniki, budując nowe języki. W HTML wszystkie znaczniki są narzucone. XML jest metajęzykiem znaczników, gdyż pozwala tworzyć nowe języki znacznikowe.

XML jest popularny z wielu powodów. Najważniejszym z nich jest łatwość obsługi i wymiany danych za pośrednictwem XML.

W XML dane i znaczniki przechowywane są w postaci tekstu, którego postać można określić. Do tworzenia dokumentów XML można używać edytorów XML. Dane nie są też kodowane w żaden sposób objęte patentami czy innymi ograniczeniami, więc są łatwiej dostępne.

Dokumenty XML same się opisują. Wykorzystując nazwy nadane poszczególnym elementom, możemy się domyślić, jakie dane zawierają poszczególne fragmenty dokumentu. Oznacza to, że dokumenty XML w znacznej mierze same się dokumentują (niezależnie od tego możliwe jest wstawianie do plików XML komentarzy).

Kolejną zaletą XML jest możliwość określenia nie tylko danych, ale także ich struktury. Jest to ważne szczególnie wtedy, gdy mamy do czynienia ze złożonymi, ważnymi danymi. Można na przykład długą transakcję bankową zapisać, jako HTML, ale w XML można także zapisać reguły semantyczne opisujące strukturę dokumentu, aby można było sprawdzić poprawność takiego dokumentu.

W XML na poprawność dokumentów kładziony jest duży nacisk. Przeglądarki XML przeprowadzają dwie kontrole: pierwsza polega na sprawdzeniu, czy dokument jest poprawnie sformułowany, druga kontrola nazywana jest walidacją.

Dokument, który jest poprawnie sformułowany musi spełniać wymagania składniowe stawiane przez utworzoną przez W3C specyfikację XML 1.0. Każdy element musi być całkowicie zamknięty w elementach nadrzędnych względem niego.

Większość przeglądarek sprawdza, czy dokumenty są poprawnie sformułowane, niektóre natomiast przeprowadzają jeszcze

walidację. Dokument XML można walidować, jeśli związana jest z nim definicja typu dokumentu (DTD) i kiedy dokument jest z nią zgodny. DTD dokumentu określa jego prawidłową składnię.

Do określenia wyglądu dokumentu XML można wykorzystać dwa narzędzia: arkusze CSS lub XSL. Standard CSS używany jest z HTML i obsługiwany przez liczne narzędzia. Za jego pomocą można określić formatowanie poszczególnych elementów, stworzyć klasy stylów, definiować czcionki, wybierać kolory, a nawet określać rozmieszczenie elementów na stronie.

Z kolei XSL jest zdecydowanie lepszy do obsługi XML, gdyż jest znacznie silniejszym narzędziem (zresztą same arkusze XSL są poprawnie sformatowanymi dokumentami XML). Dokumenty XSL składają się z reguł dotyczących dokumentów XML. Jeśli wzorzec reguły XSL pasuje do elementu XML, reguła ta przekształca dopasowany fragment kodu na inny. W ten sposób można nawet przekształcić kod XML na HTML.

Znaczniki w dokumencie określają jego strukturę. Są to:

- znaczniki początkowe
- znaczniki końcowe
- znaczniki elementów pustych
- odwołania do encji
- odwołania do encji znakowych
- komentarze
- ograniczniki sekcji CDATA
- deklaracje typu dokumentu
- instrukcje przetwarzania.

Zatem dane znakowe w dokumencie XML to wszystkie napisy niebędące znacznikami.

Strukturę dokumentu XML określa się za pomocą znaczników wyznaczających elementy. Element XML składa się ze znacznika początkowego, znacznika końcowego oraz treści, czyli tego, co się między tymi znacznikami znajduje. Wyjątkiem są elementy puste, które mogą składać się tylko z jednego znacznika o specyficznej konstrukcji.

Poprawnie sformułowany dokument XML musi zawierać jeden element, który będzie zawierał wszystkie inne elementy – jest to element główny. Element ten jest w dokumentach XML bardzo ważny, szczególnie z punktu widzenia programisty, gdyż parsowanie zawsze zaczyna się od niego.

Zgodnie ze specyfikacją XML 1.0 nazwy atrybutów podlegają tym samym regułom, które dotyczą nazw elementów.

Znaczniki zawsze są tekstem, więc i wartości atrybutów także są tekstem. Nawet,

jeśli atrybutowi przypisana jest liczba, to będzie ona traktowana, jako napis, który należy podawać w cudzysłowie.

Jednym z wymagań specyfikacji XML jest to, że nazwa atrybutu nie może pojawić się więcej niż raz w jednym znaczniku początkowym, lub znaczniku elementu pustego. W XML istnieje pięć predefiniowanych odwołań do encji. Odwołanie takie jest umieszczane tam, gdzie wystąpić ma w przetwarzanym dokumencie sama encja.

W XML istnieje pięć predefiniowanych encji ogólnych podmienianych podczas parsowania dokumentu na odpowiednie znaki:

- & - znak &
- < - znak mniejszości <
- > - znak większości >
- ' - apostrof '
- " - cudzysłów podwójny prosty ''

W XML wprawdzie jest tylko pięć predefiniowanych encji, ale można definiować nowe. Dokonuje się tego wykorzystując DTD.

W XML łatwo można definiować nowe znaczniki, jednak w miarę powstawania coraz liczniejszych aplikacji, pojawiać się będzie problem nieprzewidziany przez twórców pierwotnej wersji specyfikacji XML: konflikty nazw znaczników.

Rozwiązaniem tego problemu są przestrzenie nazw. Umożliwiają one uniknięcie konfliktów między poszczególnymi zbiorami znaczników.

7. HL7

Standard opracowywany przez grupę wolontariuszy skupionych w Health Level Seven (HL7) obejmuje swym zakresem wymianę informacji wewnątrzszpitalnej. Profesjonalnie przygotowane opracowanie standardu, przy niezbyt długim okresie funkcjonowania wskazuje na HL7 jako wzorcową grupę standaryzującą. Nazwa grupy związana jest z modelem Open System Interconnection (OSI) opracowanym przez International Standard Organization (ISO) – poziom 7.

Standard HL7 obejmuje swym zakresem definicje komunikatów przesyłanych pomiędzy poszczególnymi modułami systemu szpitalnego wskutek zaistnienia zdarzenia, o którym powinny zostać poinformowane pozostałe moduły.

Poszczególne podrozdziały dokumentu opisują następujące zagadnienia:

- ogólna struktura i zasady dotyczące przesyłanych komunikatów w standardzie HL7

- przyjęcie pacjenta, wypisanie, przeniesienie oraz rejestrację
- obsługa zleceń
- system rozliczania kosztów
- obserwacja kliniczna pacjenta
- ogólny interfejs służący do synchronizacji wspólnych zbiorów danych, np. zbiór lekarzy, zestawów badań, testów laboratoryjnych, etc.

Przy opracowywaniu zestawu wspólnych komunikatów grupa Health Level Seven założyła następujące cele:

- standard powinien wspomagać wymianę danych bez względu na wykorzystywane środowisko techniczne
- standard powinien wspomagać zarówno pełen (teoretycznie) model OSI do 7. poziomu jak i prymitywną wymianę na poziomie połączenia „punkt do punktu” za pomocą złącza RS-232C
- natychmiastowy transfer pojedynczych transakcji musi być wspomagany plikiem transferującym wiele transakcji
- standard powinien dawać możliwość uwzględniania specyfiki danego miejsca (segment typu Z w standardzie HL7)
- opracowane rozwiązanie nie może blokować naturalnego wzrostu wymagań środowiska medycznego
- standard powinien być opracowany zgodnie z obowiązującymi i występującymi na rynku protokołami transmisji bez szczególnego faworyzowania jakiegoś rozwiązania
- priorytetowym działaniem jest współpraca z innymi grupami standaryzacyjnymi np. ACR/NEMA DICOM, ASC X12, ASTM, IEEE/MEDIX, NCPDP na bazie grupy ANSI HISPP (Health Information Systems Planning Panel).

Można przedstawić następujący paradygmat odpowiedzi (w procesie wymiany uczestniczą aplikacje A i B):

- aplikacja A wysyła komunikat do B, który bez problemów odbiera go, o czym powiadamia moduł wysyłający A
- aplikacja B, po otrzymaniu komunikatu od A, stwierdza błąd aplikacji, o czym powiadamia moduł A
- aplikacja B może odrzucić komunikat, o czym także powiadamia bezzwłocznie moduł wysyłający komunikat A
- moduł odbierający B może wstrzymać wykorzystanie komunikatu od A, wysyła wówczas dwa powiadomienia, jedno o fakcie wstrzymania, a drugie o fizycznym wykorzystaniu komunikatu.

Z pojęciem przesyłanych komunikatów związane są tzw. abstrakty, które są zapisem transakcji pomiędzy modułami. Abstrakty zawierają specyfikację typu przesyłanych informacji, kiedy zostały wysłane oraz wyspecyfikowane zostają warunki wystąpienia błędów.

Informacją nadrzędną jest komunikat, który jest:

- zestawem danych transferowanych pomiędzy systemami
- każdy komunikat jest grupą złożoną z segmentów w ściśle określonym porządku
- każdy komunikat posiada swój trzyznakowy identyfikator określający jego typ.

Zdefiniowane są typy komunikatów, np.:

- ACK General acknowledgement message (ogólne potwierdzenie)
- ADT message (komunikaty przyjęciowo – wypisowe)
- ARD Ancillary RPT (display) (podrzędny RPT)
- BAR Add/change billing account (dodanie/zmiana konta rachunku)
- DFT Detail financial transaction (transakcja finansowa)
- DSR Display response (podanie odpowiedzi)
- MCF Delayed acknowledgement (opóźnione potwierdzenie)
- ORF Observation result/record resp. (wynik/zapis badania)
- ORM Order message (komunikat zlecenia),
- ORR Order acknowledgement message (potwierdzenie zlecenia)
- ORU Observation result/unsolicited (wynik dobrowolnego badania)
- OSQ Order status query (kolejka stanu zleceń)
- RAR Pharmacy Administration information (apteka – administracja)
- RAS Pharmacy Administration message (apteka – komunikaty)
- RDE Pharmacy encoded order message (apteka – zlecenia)
- QRY Query (kolejka)
- UDM Unsolicited display message (dobrowolne komunikaty).

Segmentem nazywany jest logicznie pogrupowany zestaw pól danych, który:

- może być opcjonalny lub obligatoryjny
- może wystąpić jeden raz lub wiele (dopuszczalne powtórzenia)
- każdy segment ma swój trzyznakowy identyfikator, np. dla typu ADT:
 - MSH nagłówek komunikatu
 - EVN typ zdarzenia

- PID identyfikator pacjenta.

Każdy segment składa się z atrybutów (pól), które są:

- uporządkowanym łańcuchem (ciągim) znaków
- o każdym polu wysyłane są następujące (przykładowo) informacje:
 - położenie w segmencie
 - unikalna, globalna nazwa
 - identyfikator (small integer)
 - maksymalna długość pola
 - opcjonalność
 - powtarzalność.

Standard HL7 definiuje zdarzenia związane z ADT, które generują przepływ komunikatów, np.:

- A01 Przyjęcie pacjenta
- A02 Przeniesienie pacjenta
- A03 Wypisanie pacjenta
- A04 Rejestracja pacjenta
- ...
- A28 Dodanie informacji personalnych pacjenta
- ...
- A37 Rozłączenia połączenia.

Postać komunikatu zaprezentowana została na przykładzie zdarzenia o kodzie A01 (przyjęcie pacjenta):

```
MSH nagłówek komunikatu
EVN typ zdarzenia
PID identyfikacja pacjenta
[ { NK1 } ] najbliższy krewny
PV1 wizyta pacjenta
[ PV2 ] wizyta pacjenta – uzupełnienia
[ { OBX } ] informacje o stanie zdrowia
[ { AL1 } ] alergie
[ { DG1 } ] diagnoza
[ { PR1 } ] procedury
[ { GT1 } ] dane poręczyciela
[
{ IN1 informacje o ubezpieczeniu
[ IN2 ] informacje o ubezpieczeniu
[ IN3 ] informacje o ubezpieczeniu
}
]
[ ACC ] dane o wypadku
[ UB1 ] dane kosztowe – rachunki
[ UB2 ] dane kosztowe – rachunki
```

Do prezentacji użyta została notacja BNF, w której jeżeli identyfikator występuje w nawiasach kwadratowych, wówczas jego wystąpienie jest opcjonalne, gdy w klamrowych może wystąpić jedno lub więcej powtórzeń, bez nawiasów – musi wystąpić).

Każdy z wykorzystanych w zdarzeniu A01 identyfikatorów kryje w sobie segment danych,

np. segment PID wykorzystywany przez wszystkie aplikacje, jako podstawa do transferu danych o pacjencie składa się z atrybutów (pól). Atrybut opisany jest przez informacje o jego długości, typie, ewentualnych ograniczeniach, co do czytania, może składać się z sekwencji pojedynczych pól oddzielonych separatorami.

W ramach segmentu PID występuje atrybut: Patient Name (PN) o wewnętrznym kodzie 00108, który składa się z następujących elementów:

<nazwisko>^<imię>^<wewnętrzny inicjał lub imię>^<suffix>^ prefix>^ <stopień naukowy>

Przykład komunikatu o przyjęciu pacjenta na oddział (zdarzenie A01) w standardzie Health Level Seven:

```
MSH|^~\&|REGADT|MCM|LABADT|MCM|199512171205|SECURITY|ADT^A01|MSG00001|P|2.2|<cr>
```

```
EVN|01|199512171205||<cr>
```

```
PID|||PATID1234^5^M11||MALINOWSKI^JAN^A^JR||19610615|M||C|1200 N ELM STREET^ ...GREENSBORO^NC^274011020|GL|(919)379 1212|(919)271 3434||S||...PATID12345001^2^M10|123456789|987654^NC|<cr>
```

```
NK1|KOWALSKA^ANNA^K|WIFE|<cr>
```

```
PV1|1|I|2000^2012^01|||004777^NOWAK^KAZIMIERZ^J.||SUR|||ADM|A0|<cr>
```

co oznacza:

„Pacjent Jan A. Malinowski, Jr. został przyjęty 17 grudnia 1995 roku, o godzinie 12.05 przez lekarza Kazimierza Nowaka (o identyfikatorze 004777) na oddział chirurgiczny (SUR). Został umieszczony w pokoju nr 2012, na łóżku 01, na odcinku pielęgniarskim 2000”.

Istotnym elementem każdego standardu wymiany międzymodułowej jest jego warstwa transportowa, na którą w uproszczeniu nałożono następujące warunki:

- wyeliminowanie błędów podczas transportu
- gwarancja konwersji typów kodowania
- zagwarantowana dostępność komunikatów o nieograniczonej długości.

Standard HL7 określa trzy poziomy transportowe w zależności od tego, w jakim środowisku został on zaimplementowany. Do tej pory analizowany był poziom komunikatów (najwyższy), poniżej jest poziom zasad kodowania (ang. encoding rules) a najniższy to poziom protokołu (ang. lower level protocol).

Warstwa komunikatów HL7 została opisana powyżej, warto dodać listę typów danych wykorzystywanych przez standard:

- ST łańcuch
- TX dane tekstowe
- FT formatowany tekst

- NM wartości numeryczne
- DT data
- TM czas
- TS symbol czasu
- PN pacjent
- TN telefon
- AD adres
- CK złożony identyfikator (z liczbą kontrolną)
- CN złożony identyfikator (numer + nazwa),
- CE zakodowany element
- RP wskaźnik.

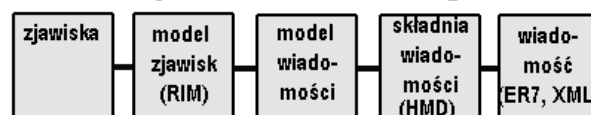
Dodatek A Standardu HL7 dokładnie prezentuje listę wszystkich wykorzystywanych słowników.

Zasady kodowania komunikatów Standardu HL7 określają:

- zasady kodowania komunikatów do postaci ciągu znaków ASCII
- zasady dekodowania komunikatu przez aplikację odbierającą
- zmienną długość pól bazującą na znakach specjalnych.

HL7 3.0 wprowadza szereg modeli jednostek i aktywności występujących w służbie zdrowia i na tej podstawie odwzorowuje je na wiadomości. Podstawowym, opracowanym przez HL7, referencyjnym modelem jest obiektowy model informacji w służbie zdrowia o nazwie RIM (ang. Reference Information Model). Model ten przez zastosowanie języka modelowania obiektowego UML przedstawia 123 klasy obiektów, możliwe stany oraz relacje pomiędzy obiektami. Uproszczeniem tego modelu jest model informacyjny wiadomości – MIM (ang. Message Information Model) ujmujący te klasy, których obiekty są odwzorowywane na wiadomości. Dla każdego obiektu można prześledzić trasę jego powiązań poczynając od obiektu-korzenia. Wszystkie relacje dla danego typu obiektu tworzą określone drzewo (diagram) atrybutów i stanów związanych z modelowanym zjawiskiem (np. z pacjentem). Stworzony w ten sposób diagram jest określanym mianem MOD (ang. Message Object Diagram). Bazując na diagramie tworzony jest hierarchiczny zapis atrybutów i relacji obiektu (HMD – Hierarchical Message Description), który jest wprost rzutowany na elementy wiadomości.

Generacja wiadomości według HL7 3.0



Rys. 9. Proces formowania wiadomości w HL7 wersja 3.0

Cały proces od określenia potrzeb i stanu zjawisk poprzez modelowanie do implementacji obiektów w formie wiadomości określany jest w HL7 MDF (Message Development Framework). HL7 opracowując wersje 3.0 pragnie powiązać organizację gromadzenia i reprezentacji danych w służbie zdrowia z wymianą danych. Opracowany model referencyjny RIM ma więc charakter uniwersalny i może być wykorzystywany w tworzeniu systemów informacyjnych służby zdrowia. Co więcej, samo kodowanie treści wiadomości może odbywać się za pomocą różnych metod, jak np. ER7 (encoding rules 7), XML (eXtended Markup Language) czy EDIFACT.

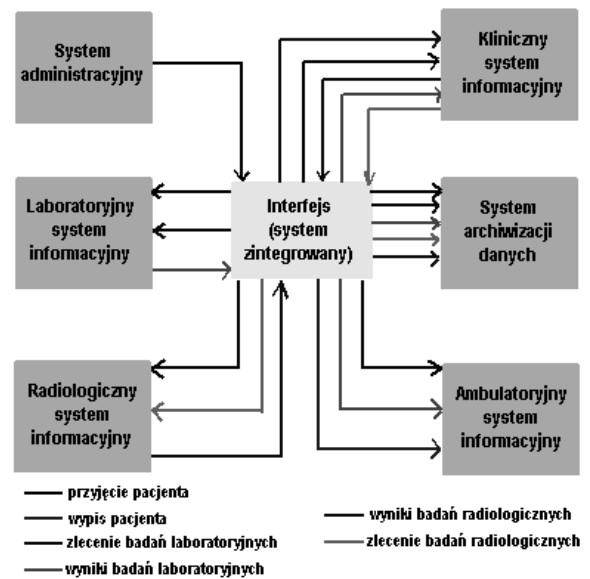
Struktura wiadomości posiada odmienną formę od tych formowanych do wersji 2.3.1 włącznie. Pojawiają się zupełnie nowe segmenty (np. ENC – encounter – spotkanie) oraz oznaczania list (BL – begin list – początek listy, EL – end list – koniec listy) grup (BG – begin group – początek grupy, EG – end group – koniec grupy) i innych elementów składni.

Znaczny wysiłek prac w grupie HL7 jest skierowany na stworzenie implementacji wiadomości opracowywanej nowej normy HL7 3.0 w formie dokumentów XML. Możliwe do stworzenia definicje typów dokumentów (DTD – Document Type Definitions) stanowią oddzielne względem plików XML formy dokumentów. Możliwe jest stworzenie różnych form dokumentów np. dla różnych wiadomości.

Wraz z opracowywaniem nowego standardu wiadomości, modelu referencyjnego oraz typów dokumentów XML dla wiadomości HL7 postanowiło opracować architekturę elektronicznych kart pacjenta. Bazując na modelu referencyjnym RIM opracowano szereg form dokumentów DTD oraz procedur ich wymiany. Przedsięwzięcie to nazwano PRA – Patient Record Architecture. Ponieważ dokumenty PRA są implementacją XML stanowią opracowania niezależne od urzędzeń, czy rozwiązań sprzętowo-programowych.

HL7 wersja 3.0, formy dokumentów DTD, PRA i inne opracowania związane z nowym sposobem reprezentacji zjawisk w medycynie są przygotowane i czekają na ostateczne ich zaakceptowanie jako norm. Istniejące i tworzone rozwiązania Szpitalnych Systemów Informacyjnych umożliwiać mogą różnorodne formatowanie wiadomości w procesie ich wymiany np. zgodnie z HL7 2.x, albo HL7 3.0 lub w formie PRA. Od możliwości interpretacji wymienianych wiadomości (interfejsy) zależy

więc funkcjonalność całego systemu informacyjnego.



Rys. 10. Przykład przepływu wiadomości między systemami informacyjnymi służby zdrowia.

8. Podsumowanie

BPMN został opracowany do realizacji dwóch celów. Po pierwsze jest łatwy w zrozumieniu i stosowaniu. Można go wykorzystać do szybkiego modelowania procesów biznesowych i jest łatwy w zrozumieniu dla użytkowników pozbawionych umiejętności technicznych. Po drugie dostarcza możliwości modelowania skomplikowanych i złożonych procesów biznesowych i może być bez problemu przełożony na dowolny język wykonawczy procesów biznesowych. Istnieje możliwość przekształcenia modelu BPMN na model UML i wykorzystania go do budowy systemu informatycznego.

W kontekście architektury systemu informatycznego za pomocą języka UML modelujemy zarówno statyczny jak i dynamiczny aspekt systemu.

Z wykorzystaniem języka UML modelujemy złożone systemy za pomocą małego zbioru prawie niezależnych modeli. Tworzymy modele, które mogą być konstruowane i przetwarzane oddzielnie, ale są nadal wzajemnie powiązane.

Aby zrozumieć architekturę systemów zorientowanych obiektowo, potrzebujemy połączyć kilka uzupełniających się widoków: widok przypadków użycia, widok logiczny, widok procesów, widok implementacyjny, widok wdrożeniowy.

UML jest używany także do modelowania procesów biznesowych i reprezentowania struktur organizacyjnych.

Za pomocą języka wyrażeń OCL możemy uszczegóławiać modele UML, dodając warunki początkowe, warunki końcowe i niezmienniki dla klas i metod w modelu.

Język XML jest standardem, który w dużym stopniu stał się medium dla opisywania, komunikowania i realizowania strategii zarządzania informacją.

XML jest popularny z wielu powodów. Najważniejsza z nich jest łatwość obsługi i wymiany danych za pośrednictwem XML. Język XML wykorzystywany jest do strukturyzacji danych. Tak przygotowane dane mogą być zapisywane w postaci plików (dokumentów) i wykorzystywane w innych systemach informatycznych. XML możemy także zastosować do strukturyzacji danych wysyłanych w komunikatach między systemami informatycznymi lub ich modułami.

HL7 w wersji 3.0 to zarówno standard kodowania wiadomości (także w języku XML) jak i model referencyjny RIM i dokumenty PRA. Opracowany model referencyjny RIM ma charakter uniwersalny i może być wykorzystywany w tworzeniu systemów informacyjnych służby zdrowia. Wraz z opracowywaniem nowego standardu wiadomości, modelu referencyjnego oraz typów dokumentów XML dla wiadomości, HL7 opracowało architekturę „elektronicznych kart pacjenta”. Bazując na modelu referencyjnym RIM opracowano szereg form dokumentów DTD oraz procedur ich wymiany. Przedsięwzięcie to nazwano PRA – Patient Record Architecture. Ponieważ dokumenty PRA są implementacją XML stanowią opracowania niezależne od urzędzeń, czy rozwiązań sprzętowo-programowych.

Język GELLO może być używany do budowania zapytań do wydobywania i manipulowania danymi z historii choroby w systemach służby zdrowia. GELLO służy do konstruowania kryteriów decyzyjnych przez tworzenie wyrażeń dla szczególnych danych cecha/wartość. Te kryteria mogą być używane w bazach wiedzy systemów wspomaganie decyzji takich jak te, zaprojektowane by dostarczyć ostrzeżeń i przypomnień, wskazówek, albo innych zasad decyzyjnych. GELLO jest specjalizacją języka OCL, dzięki czemu można go wykorzystać do uszczegóławiania modeli UML. GELLO wykorzystuje model referencyjny RIM standardu HL7.

Po analizie rozpatrywanych standardów celowym wydaje się zastosowanie UML, XML, HL7 oraz GELLO do modelowania architektury systemu informatycznego dla służby zdrowia.

9. Bibliografia

- [1] Fowler M., *UML Distilled 3rd Edition*, Addison-Wesley, Kanada, 2005.
- [2] Johnston S., *UML 2.0 Profile for Software Services*, 2005, (www.ibm.com).
- [3] Johnston S. *Rational UML Profile for Business Modeling*, 2004, (www.ibm.com).
- [4] Rozanski N., *Software Systems Architecture*, Pearson Education Inc., Crawfordsville, USA, 2005.
- [5] Warmer J., Kleppe A., *OCL precyzyjne modelowanie w UML*, WNT, Warszawa, 2003.
- [6] Holzner S., *XML Vademecum profesjonalisty*, Helion, Gliwice, 2001.
- [7] Pender T., *UML Bible*, Wiley Publishing, Inc., Indianapolis, 2003.
- [8] *Business Process Modeling Notation version 1.2*, Object Management Group, 2009, (www.omg.org/spec/BPMN/1.2).
- [9] Sordo M., Ogunyemi O., Boxwala A., Greenes R., *Software Specifications for GELLO: An Object-Oriented Query and Expression Language for Clinical Decision Support*, Decision Systems Group, Harvard Medical School, 2003.

Przydatność języka Gello i standardu RIM do obiektowego modelowania systemów medycznych

P. KOSIUCZENKO

e-mail: piotr.kosiuczenko@wat.edu.pl

Instytut Systemów Informatycznych
Wydział Cybernetyki WAT
ul. Kaliskiego 2, 00-908 Warszawa

Współczesne systemy medyczne charakteryzują się dużą złożonością i heterogenicznością. Wymaga to ich ujednoczenia, także za pomocą standardów. Health Level 7 (HL7) jest szeroko zaakceptowanym standardem pozwalającym na wymianę danych medycznych pomiędzy różnymi instytucjami i systemami informatycznymi. Stowarzyszony z nim standard Reference Information Model (RIM) dotyczy składni i semantyki wiadomości przesyłanych pomiędzy różnymi systemami medycznymi. Tak jak diagramy klas języka UML, modele oparte o RIM wymagają uszczegółowienia. W tym celu powstał klon języka OCL zwany Gello.

W tej pracy scharakteryzowane zostały wyżej wspomniane standardy oraz powiązania pomiędzy nimi. Zbadana została także ich przydatność do modelowania i projektowania systemów medycznych.

Słowa kluczowe: HL7, RIM, Gello, UML, medycyna, ścieżki kliniczne

1. Wprowadzenie

System opieki zdrowotnej, podobnie jak wiele innych systemów istotnych dla społeczeństwa, jest w swojej naturze bardzo złożony i wymaga właściwego wsparcia ze strony informatyki. Procesy diagnozowania i leczenia wiążą się z wytwarzaniem i przesyłaniem różnorodnych dokumentów. Mogą one być bardzo skomplikowane ze względu na różnorodność terminologii medycznej oraz posiadać złożoną strukturę. W związku z tym powinny one mieć zestandaryzowaną formę. Potrzebne są tu standardy dotyczące struktury, znakowania i semantyki wymienianych dokumentów w celu zapewnienia interoperacyjności systemów informatycznych. Heterogeniczne systemy używane dziś w medycynie często nie są w stanie tego zapewnić. Potrzebny jest też standard wymiany dokumentów.

Najbardziej rozpowszechnionym standardem w tej dziedzinie jest Health Level 7 (HL7) [2] i stowarzyszony z nim standard Reference Information Model (RIM) [3] (zob. także [10]). HL7 jest standardem ANSI dotyczącym wymiany danych pomiędzy systemami informatycznymi opieki zdrowotnej. Definiuje on gramatykę i słownictwo dokumentów, tak aby umożliwić ich wymianę pomiędzy heterogenicznymi systemami opieki zdrowotnej. Standard RIM jest obiektowo zorientowany. Definiuje on składnię i semantykę przesyłanych wiadomości. RIM jest powiązany

z językiem UML [6] będącym standardem do modelowania systemów obiektowo zorientowanych.

Język UML jest zdefiniowany na meta poziomie poprzez tak zwany meta-model będący pewnym szczególnym diagramem klas. Diagramy klas, jako takie, nie posiadają wystarczającej ekspresywności i muszą być uszczegółowione poprzez dodatkowe wymagania. Do tego celu używany jest język Object Constraint Language (OCL) [5]. Jego odpowiednikiem w przypadku HL7 i RIM jest Gello [9].

W tym artykule omawiana jest przydatność standardu RIM i języka Gello do obiektowego modelowania repozytoriów medycznych. RIM i Gello porównane są z językami UML i OCL.

2. Charakterystyka HL7 i RIM

Health Level 7 (HL7, [2]) jest standardem służącym do wymiany danych medycznych i opisu tzw. workflows związanych z diagnozowaniem i leczeniem. Ma on zapewnić interoperacyjność pomiędzy różnymi systemami informatycznymi używanymi w opiece medycznej. Liczba 7 odnosi się do siódmej, najwyższej warstwy w ISO/OSI protokole przesyłu danych. Zwana jest ona „warstwą aplikacji”. Najniższa warstwa zwana jest warstwą fizyczną i dotyczy wymiany danych na poziomie maszyn i bitów. Najnowsza wersja standardu ma numer 3 (oznaczana jest tu jako

HL7 V3). W przeciwieństwie do wersji numer 2, jest ona oparta na formalnej, obiektowo zorientowanej metodologii. HL7 V3 jest też znacznie bardziej dopracowanym dokumentem. Warto tu zaznaczyć, że istnieje polska wersja interfejsu do wymiany danych pomiędzy szpitalami wyspecyfikowana zgodnie ze standardem HL7 (zob. [4]).

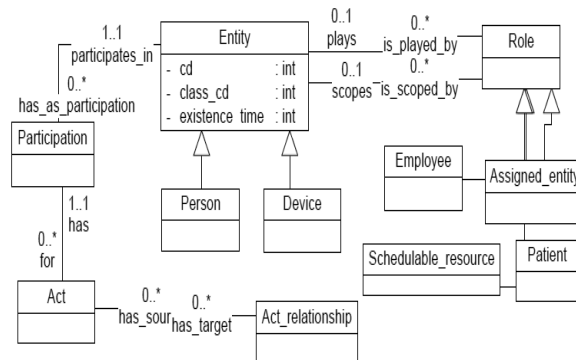
HL7 V3 jest oparty na tzw. Reference Information Model (RIM). RIM jest obiektowo zorientowanym meta-modelem pozwalającym na specyfikację danych (np. danych laboratoryjnych, danych klinicznych, wyników badań klinicznych itp.), a szczególnie wiadomości, zarówno pod względem ich gramatyki, jak i semantyki, w celu zapewnienia interoperacyjności systemów medycznych (zob. np. [1]).

RIM służy też do specyfikacji takich elementów systemów medycznych, jak procesy i workflows. Kręgosłupem RIM są klasy Act, Role i Entity. Są one połączone ze sobą za pomocą asocjacji będących jednocześnie klasami (ang. *association class*), takimi jak: Act_relationship, Participation i Relationship_link (zobacz np. rys. 1).

Klasa Act odgrywa fundamentalną rolę w RIM. Jej elementy nazywane są tu działaniami (ang. *acts*). Działanie jest zamierzonym postępowaniem w dziedzinie opieki zdrowotnej. Działania mogą być powiązane ze sobą za pomocą asocjacji z klasy Act_relationship. Każde działanie występuje w pewnym modusie, który opisuje jego stan od zdefiniowanego, poprzez planowany do zakończonego. Każde działanie ma tylko jeden modus w aktualnym momencie swego cyklu życia.

Klasa Entity odpowiada jednostkom, takim jak ludzie, miejsca i rzeczy. Jej elementy, zwane tu jednostkami, są powiązane poprzez role z innymi jednostkami, a poprzez tzw. partycypacje, tj. elementy klasy Participation, biorą udział w działaniach, tj. elementach klasy Act.

Klasa Role odpowiada rolom, które mogą być odgrywane przez jednostki. Tak jak w języku UML, role są jakby kapeluszkami lub funkcjami, które mogą być „zakładane” lub przyjmowane przez jednostki.



Rys. 1. Podstawowe klasy RIM za [3]

Participation (uczestnictwo) również jest jedną z podstawowych klas. Określa ona, w jaki sposób obiekt z klasy Entity, odgrywający pewną specyficzną rolę, funkcjonuje w zakresie działania. Uczestnictwo jest ograniczone do zakresu działania, w przeciwieństwie do roli określającej kompetencje jednostki niezależnie od jakiegokolwiek działania. Podmioty mogą uczestniczyć w działaniu na wiele sposobów. Na przykład osoby w roli Certified_practitioner mogą uczestniczyć w działaniu będącym elementem klasy Patient_encounter jako lekarze.

Podobnie jak UML, RIM definiuje pojęcie aktora. Aktorzy mogą uczestniczyć w działaniach na różne sposoby. Na przykład dentysta i technik dentystyczny są podmiotami w zabiegu dentystycznym. Są oni mniej lub bardziej bezpośrednio zaangażowani w zabieg. Pacjent może być sam aktorem wykonującym samodzielnie zabieg (np. dezynfekcja i bandażowanie rany, czy auto-iniekcja insuliny)

RIM różni się od UML przede wszystkim tym, że z jednej strony nie posiada tak rozbudowanego meta-modelu i zbioru diagramów, zaś z drugiej strony posiada liczny zbiór dedykowanych klas, które są przeznaczone do modelowania różnych artefaktów medycznych. Zamiast ograniczeń zapisanych w języku naturalnym lub OCL [5], używane są tu tak zwane diagramy blokowe (ang. *Block Diagrams*). Zdaniem autorów standardu RIM (zob. [7]) są one prostsze w użyciu niż OCL, czy komentarze w języku naturalnym umieszczone w prostokątach połączonych z elementami modelu, jak ma to miejsce w przypadku UML-a.

Z czasem zauważono jednak, że niezależnie od tego czy diagramy blokowe są naprawdę prostsze w użyciu, czy też nie, nie wystarczają one do specyfikacji wielu aspektów. Ogólnie rzecz biorąc tego rodzaju diagramy nie mają

odpowiedniej ekspresywności i wiele istotnych ograniczeń nie może być wyrażonych za ich pomocą. Odpowiedzią na te problemy było powstanie języka Gello [9].

3. Gello versus OCL

Wiele ograniczeń dotyczących obiektów i powiązań między nimi nie może być wyrażonych za pomocą samych diagramów klas. Z drugiej strony diagramy klas nie nadają się do specyfikacji metod. Nie są one w stanie wyrazić warunków, które muszą być spełnione podczas funkcjonowania systemu. Jest to możliwe dopiero za pomocą tzw. kontraktów. Potrzebne są formuły wyrażające warunki, kiedy metoda może być wykonana i jaki jest jej rezultat oraz warunki, które muszą być zawsze spełnione.

Są trzy rodzaje formuł tworzących kontrakt. Warunki wstępne (*pre-conditions* w terminologii OCL) pozwalają opisać kiedy metoda może być wywołana, a w szczególności nakładają one ograniczenia na aktualne parametry metody. Warunki końcowe (*post-conditions* w terminologii OCL) pozwalają opisać stan w jakim się ma znaleźć system po wykonaniu metody oraz relację pomiędzy nim a stanem poprzedzającym wykonanie metody. Niezmienniki (*invariants* w terminologii OCL) pozwalają na wyrażenie warunków spójności, które muszą być spełnione podczas działania systemu. OCL jest czystym językiem zapytań, tj. sprawdzenie warunku z założenia nie zmienia stanu obiektów. OCL jest językiem o silnej typizacji wyrażań, tj. każde wyrażenie tego języka musi mieć swój typ.

Gello jest wersją języka OCL skonstruowaną specjalnie dla standardu RIM [3]. W 2005 roku został on zaakceptowany jako standard ANSI pod nazwą HL7 Gello. W zasadzie jest on podzbiorem OCL-a. Tak jak OCL jest to czysty język zapytań, który pozwala na kontraktową specyfikację systemów. Typy danych są zdefiniowane podobnie jak w OCL, jednak w odróżnieniu od OCL-a nie ma tu automatycznego spłaszczania kolekcji składających się z innych kolekcji. Kolekcja kolekcji może być spłaszczona za pomocą specjalnego operatora *flatten*.

4. CDA

Clinical Document Architecture (CDA) [2] jest standardem służącym do znakowania dokumentów klinicznych, specyfikacji ich struktury i semantyki. Jest to część standardu HL7 V3. Jego celem jest ujednoczenie dokumentów klinicznych umożliwiające ich wymianę.

Dokument kliniczny specyfikuje obserwacje i badania pacjenta, udzielane usługi, trwałość dokumentów (tj. czas, przez który dokument musi być zachowany w niezmienionej formie), organizację odpowiedzialną za dokument, dane dotyczące wiarygodności (ang. *authentication*) i jego kontekst. Standard ten jest oparty na języku XML. Zawartość i semantyka dokumentów są zdefiniowane za pomocą standardu RIM. Dane są zapożyczone ze standardu HL7 V3. Dokumenty CDA mogą być przesyłane w wiadomościach zgodnych ze standardem HL7 V3.

5. Ewaluacja RIM i Gello

HL7 i w szczególności RIM ma swoich krytyków, którzy podważają celowość rozwijania tego standardu, jak na przykład B. Smith, W. Ceusters w pracy [8]. Mankamenty tego standardu wspomniani autorzy przedstawiają następująco:

- Interoperacyjność – pomimo że standard RIM został przyjęty już w 1996 roku, po ponad dziesięciu latach starań i inwestowania w RIM interoperacyjność nie została tak naprawdę osiągnięta.
- Przydatności w nowych dziedzinach zastosowań – RIM opiera się na zestawie „normatywnych” klas (takich jak wyżej wspomniane *Act*, *Role* i *Participation*), z których każda posiada liczny zbiór odpowiednio zinterpretowanych atrybutów. Gdy RIM jest stosowany do nowej dziedziny (jak np. farmaceutyka), należy wybrać tylko pewne z nich. Nie jest jednak jasne, czy wybór atrybutów we wszystkich przypadkach wystarcza.
- Zakres – z klas podstawowych tylko klasa *Entity* może być użyta do modelowania tych rzeczy, które nie są działaniami. Jednak nie wiadomo, jak na tej podstawie można w przejrzysty sposób użyć RIM do modelowania np. aspektów związanych z procesem choroby, czy interakcji pomiędzy lekami, gdyż te ostatnie nie są ani jednostkami, ani działaniami.
- Dokumentacja – dokumentacja oparta na RIM jest niejasna i słabo zintegrowana z innymi częściami składowymi HL7 V3, dla których RIM jest podstawą. Ma ona wiele wewnętrznych niespójności, na przykład w przypadku użycia takich pojęć, jak działanie, instancja klasy *Act*, obiekt klasy *Act*. Jest to szczególnie niepożądane

w przypadku standardów dotyczących wiadomości.

- Trudność nauczenia się – HL7 V3 jest trudny do opanowania dla szerokiego grona użytkowników ze względu na raczej amatorską dokumentację, dużą liczbę przypadków specjalnych, częste zmiany i uzupełnienia oraz trudności w definicji wiadomości.
- Marketing – wiele spośród marketingowych haseł propagujących HL7 V3 jako „standard danych dla biomedycznej informatyki” nie jest uzasadnione, biorąc pod uwagę, że wciąż istnieje wiele nierozwiązanych problemów technicznych.

W pracy [7] przedstawiona jest odpowiedź głównych projektantów RIM na krytykę tego standardu. Twierdzą oni, że krytyka standardu jest niekonstruktynwa i przypominają główne założenia RIM, nie wdając się jednak w szczegółową dyskusję.

Trzeba tu zaznaczyć, że powyższa krytyka w punktach dotyczących interoperacyjności i marketingu nie jest czymś wyjątkowym w przypadku HL7 i RIM. Występuje ona także w przypadku wielu innych standardów. Pozostałe punkty wydają się jednak w dużej mierze trafione.

Zaletą RIM jest dopasowanie do potrzeb modelowania i projektowania systemów medycznych, w tym bardzo bogaty zbiór odpowiednich klas. Warto tu dodać, że jeśli chodzi o nowe dziedziny zastosowań, to zasadniczo można tu użyć standardowych metod rozszerzania i dopasowywania meta-modelu tak, jak są one używane w przypadku języka UML. Dokładniej możliwa jest definicja odpowiednich profili tego języka. Każdy model UML-a jest instancją (ang. *instantiation*) meta-modelu UML-a i ma się do niego tak, jak obiekt do klasy. Istnieje tu możliwość użycia stereotypów nadających pewne szczególne znaczenie konkretnym klasom (np. do oznaczenia choroby) oraz innym elementom modelu. Istnieje też możliwość wprowadzania nowych meta-klas i atrybutów, które mogą modelować konkretne klasy, odpowiadające pewnym szczególnym zastosowaniom oraz powiązania między nimi. Ta ostatnia opcja nazywana jest rozszerzeniem wagi ciężkiej, gdyż modyfikuje oryginalny meta-model UML-a.

Jednakże autorzy standardu RIM nie przewidują tego rodzaju rozszerzeń i wymagają, aby klasy wprowadzane przy specyfikacji rozszerzały klasy występujące w modelu RIM (zob. [7], rozdział drugi), co znacznie ogranicza możliwość adaptacji tego standardu. Autorzy nie

wspominają o stereotypizacji klas i ogólnie o profilach. Nie jest to też zgodne z ideą UML-a, że każdy model jest instancją meta-modelu. Zamiast tego proponowany jest mechanizm rzutowania polegający na pominięciu niepotrzebnych atrybutów – raczej na zasadzie ich nieużywania, jeśli nie są potrzebne, niż dosłownego ich usuwania z modelu – jak również poprzez dodatkowe warunki uszczegóławiające dany model. Takie podejście ma zapewnić interoperacyjność, ogranicza ono jednak znacznie możliwość adaptacji modelu RIM, a przez to także jego zastosowania.

Wspomniane diagramy blokowe mają z jednej strony umożliwić bardziej szczegółową specyfikację systemów, niż za pomocą samych diagramów klas, z drugiej strony mają być bardziej czytelne niż formuły OCL-a. Jak wspomniano w poprzednim rozdziale, diagramy blokowe w żadnym razie nie mogą zastąpić OCL-a, gdyż nie mają wystarczającej mocy opisowej. Ponadto osobie mającej pewną wprawę w czytaniu diagramów UML-a zdecydowanie łatwiej jest czytać diagramy klas z dodatkowymi ograniczeniami zapisanymi np. w języku OCL niż diagramy blokowe.

Wspomniane powyżej mankamenty nie wiążą się bezpośrednio z zastosowaniami w medycynie. Główny problemem ze standardem RIM polega na tym, że pod względem metodologicznym jest on wciąż mało dopracowany. Wiąże się to z faktem, iż powstawał w dużej mierze równoległe z językiem UML i podczas jego tworzenia niektóre istotne ustalenia i rozwiązania metodologiczne dotyczące tego ostatniego języka nie były znane.

Stowarzyszony z RIM język Gello jest klonem języka OCL, niewiele odbiegającym od swego pierwowzoru i jako taki ma mocne podstawy metodologiczne. Jeśli chodzi o stosowalność HL7, a w szczególności RIM, to kluczową rolę odgrywają tu dwa fakty. Po pierwsze powszechność użycia – jeśli standard jest szeroko przyjęty, to jego użyciu trudno się oprzeć, nawet jeśli ma poważne mankamenty natury metodologicznej. Po drugie istotne jest ulepszanie standardu. Wielu wspomnianym problemom można byłoby zaradzić poprzez przyjęcie odpowiednich zmian w standardzie.

6. Konkluzja

W niniejszej pracy omówiony został standard HL7, blisko związany z nim standard RIM, standard CDA oraz język Gello. Zaletą HL7 i RIM są bliski związek z zastosowaniami medycznymi i doświadczenie w ich użyciu.

Za przyjęciem HL7 i RIM przemawia ich popularność. Przeciw ich przyjęciu przemawiają niedoskonałości metodologiczne oraz nie-precyzyjność specyfikacji. Język Gello jest klonem języka OCL i jako taki ma dobre podstawy metodologiczne oraz teoretyczne, nie wprowadza on jednak żadnych istotnych zmian, czy adaptacji w celu specyfikacji systemów medycznych.

Mimo technicznych niedoskonałości RIM, trudno byłoby zalecać przyjęcie nowego standardu, gdyż o użyteczności standardu decydują nie tylko jego strona metodologiczna, narzędzia wspierające i formalne, ale przede wszystkim powszechność jego użycia. W przypadku kiedy nie zbierze się konsorcjum o odpowiednim potencjale finansowo-politycznym, mogące przeforsować szerokie użycie nowego standardu, jego wprowadzanie jest raczej skazane na niepowodzenie.

Inaczej ma się sprawa z Gello i OCL. Jak wspomnieliśmy, Gello jest zaledwie klonem OCL-a i właściwie stanowi jego fragment. OCL jest szeroko znany i bardzo dobrze dopracowany od strony metodologicznej. Z drugiej strony istnieją liczne narzędzia UML-a ten język wspierające, w tym szeroko stosowane, jak na przykład Rational Architect i Together. Dlatego też język OCL może być rozważony jako alternatywa dla Gello.

7. Bibliografia

- [1] B. Blobel, G. Stassinopoulos, P. Pharow, „Model-Based Design and Implementation of Secure, Interoperable EHR Systems”, *Proc. Of AMIA Annual Symposium*, 96-100, 2003.
- [2] HL7 Board, HL7, <http://www.hl7.org/>.
- [3] HL7 Board, „HL7 Reference Information Model”, V 01-12, 1.2.2002, http://www.miforum.net/distillate/rim/rim0112_body.htm.
- [4] CompuGroup, „Projekt HL7PL”, Polska Wersja Standardu HL7: http://www.uhc.com.pl/index.php?option=com_content&task=view&id=23&Itemid=75.
- [5] OMG, „Object Constraint Language”, Version 2.0, formal/06-05-01, OMG, 2006.
- [6] OMG: „Unified Modeling Language Specification”, Superstructure, Version 2.2, formal/2009-02-02, 2009.
- [7] G. Schadow, C. Mead, D. Walker, „The HL7 Reference Information Model Under Scrutiny”, *Studies in health technology and informatics*, Vol. 124, 2006, str.151-156.
- [8] B. Smith, W. Ceusters, „HL7 RIM: An Incoherent Standard”, *Studies in Health Technology and Informatics*, vol. 124, 133-138, 2006.
- [9] M. Sordo, O. Ogunyemi, A. Boxwala, R. Greene, S. Tu, „GELLO: An Object-Oriented Query and Expression Language for Clinical Decision Support”, *Technical Report*, DSG-TR-2003-02, Harvard Medical School, 2003.
- [10] Y. Sumita, M. Takata, K. Ishitsuka, Y. Tominaga, K. Ohe, „Building a reference functional model for EHR systems”, *International Journal of Medical Informatics*, vol. 76(9), 688-700, 2007.

Wielokryterialne zadanie optymalizacji schematu agregatów w hurtowniach danych

M. MAZUREK

e-mail:marcin.mazurek@wat.edu.pl

Instytut Systemów Informatycznych
Wydział Cybernetyki WAT
ul. S. Kaliskiego 2, 00-908 Warszawa

Dominującą technologią wykorzystywaną do zwiększenia wydajności dużych hurtowni danych są, wyliczane w oparciu o dane szczegółowe, agregaty, w tym mające postać wielowymiarowych kostek. Mechanizmy przepisywania zapytań pozwalają na znacznie szybsze wyznaczenie odpowiedzi na zapytanie w oparciu o mniejsze agregaty niż w przypadku odwołań do źródłowych tabel hurtowni. Do kluczowych parametrów opisujących schemat wielowymiarowych agregatów należą czas odpowiedzi na zapytania użytkownika, czas aktualizacji danych w schemacie w oparciu o nowe dane w hurtowni oraz przestrzeń dyskowa wymagana do przechowywania agregatów. Przedstawione zostało dwukryterialne sformułowanie zadania optymalizacji, w którym jako ograniczenie przyjęty został czas aktualizacji danych. Pozostałe dwie zmienne tworzą wektor kryteriów. Wyznaczony został zbiór rozwiązań optymalnych w sensie Pareto oraz zaproponowano metodę znalezienia jednoznacznego rozwiązania w oparciu o punkt idealny.

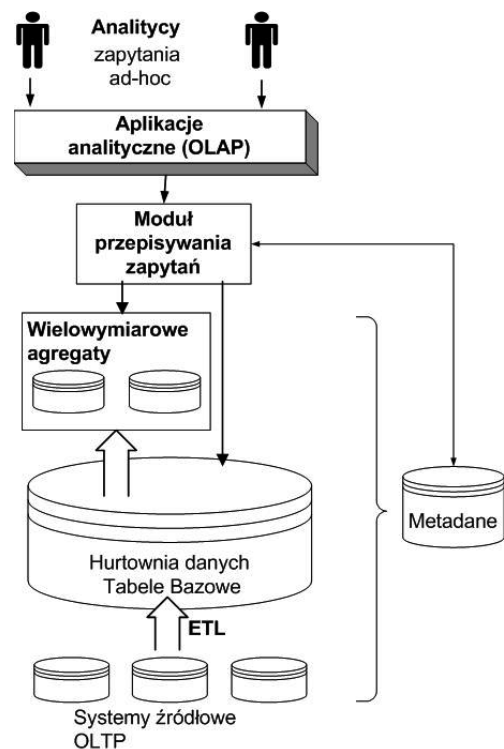
Słowa kluczowe: hurtownia danych, OLAP, wielowymiarowe agregaty, optymalizacja

1. Wprowadzenie

Wykorzystywane w systemach analizy danych klasy OLAP wielowymiarowe agregaty, zwane kostkami (ang. *cube*), umożliwiają zmniejszenie czasu odpowiedzi na zapytania użytkowników. Zarówno logiczny model danych, jak też fizyczna organizacja tych zbiorów podporządkowane są minimalizacji czasu oczekiwania na wyniki zapytań klasy SPJG (zapytania Select-Project-Join z pojedynczą instrukcją agregacji group-by na końcu [1]), formułowanych w trybie ad-hoc przez użytkowników.

Umieszczenie schematu agregatów w architekturze hurtowni danych przedstawia rys. 1. W głównej składnicy hurtowni przechowywane są dane szczegółowe. Ze względu na duży wolumen danych, wielokrotnie przekraczający rozmiar pamięci operacyjnej, każdorazowe wyliczenie odpowiedzi na zapytania użytkownika o zagregowane dane wiązałoby się ze stosunkowo długim czasem oczekiwania na odpowiedź, ze względu na dużą liczbę operacji wejścia-wyjścia. Rozwiązaniem jest wyliczenie i przechowywanie części zagregowanych danych tak, by w chwili zapytania użytkownika nie było konieczności przeprowadzania obliczeń. Jeżeli schemat agregatów nie będzie zawierał danych, o które pyta użytkownik, wówczas zapytanie zostanie przekierowane przez moduł przepisywania zapytań do tabel bazowych w głównym

repozytorium hurtowni danych. Uzupełnieniem systemu jest baza metadanych, zawierająca istotne informacje sterujące działaniem podsystemu agregatów: model danych przechowywanych w hurtowni, model agregatów oraz informacje wolumetryczne.



Rys. 1. Architektura hurtowni danych

Zagadnienia badawcze w obszarze wielowymiarowych struktur danych dla aplikacji OLAP dotyczyć mogą następujących obszarów:

- projekt schematu agregatów: wybór zakresu danych przechowywanych w postaci wielowymiarowych agregatów
- utrzymanie schematu: określenie wydajnego sposobu odświeżania danych w kostkach, po aktualizacji danych w tabelach bazowych
- wykorzystanie schematu: mechanizmy przepisania zapytań użytkownika, umożliwiające wyznaczenie odpowiedzi na zapytanie użytkownika w oparciu o schemat agregatów, bądź, jeżeli jest to niemożliwe, w oparciu o tabele bazowe.

Przedmiotem rozważań w artykule jest pierwsze z wymienionych zagadnień, przy czym pozostałe dwa muszą zostać uwzględnione w postaci kształtu odpowiednich funkcji występujących jako dane w zadaniu wyznaczenia schematu agregatów.

W literaturze problemu zaproponowano kilka sformułowań zadania optymalizacji różniących się funkcją kryterium i ograniczeniami. W praktyce, wyznaczenie schematu agregatów w oparciu o jedno, wybrane kryterium optymalizacji nie jest satysfakcjonujące dla użytkownika.

W artykule przedstawiony został model schematu agregatów (meta-model M2 zgodnie z klasyfikacją MOF [2][3]) oraz sformułowanie wielokryterialnego zadania optymalizacji. Dla wskazanego zadania wyznaczony został zbiór Pareto-optymalny oraz sposób znalezienia rozwiązania w tym zbiorze.

2. Model wymiarowy danych i schemat agregatów

Wprowadźmy oznaczenie zbioru wymiarów analizy:

$$\mathbf{D} = \{dim_1, dim_2, \dots, dim_n, \dots, dim_N\} \quad (1)$$

gdzie dim_n oznacza n -ty wymiar analizy.

W każdym z wymiarów można wyróżnić jeden lub więcej poziomów szczegółowości (atrybutów). Niech L_n oznacza liczbę poziomów agregacji wyodrębnionych dla n -tego wymiaru.

Zbiór atrybutów odpowiadających poziomom agregacji wymiarów wyznaczonym przez atrybuty określa macierz:

$$\mathbf{L} = \left\{ \begin{array}{l} attr_1^1, attr_2^1, \dots, attr_{L_1}^1, \\ attr_1^2, attr_2^2, \dots, attr_{L_2}^2, \\ \dots, \\ attr_1^N, attr_2^N, \dots, attr_{L_N}^N \end{array} \right\}. \quad (2)$$

Każdy podzbiór poziomów agregacji L_n

$$L_n = \{attr_1^n, attr_2^n, \dots, attr_{L_n}^n\} \quad (3)$$

tworzy kratę $\langle L_n, \preceq \rangle$ w oparciu o relację porządku \preceq , określającą zależność funkcyjną pomiędzy wartościami atrybutów (wartości atrybutu $attr_i^n$ wymiaru dim_n zależą funkcyjnie od wartości atrybutu $attr_j^n$), co zostało opisane w postaci definicji (4).

$$attr_i^n \preceq attr_j^n \stackrel{df}{\Leftrightarrow} \exists f : Dom(attr_j^n) \xrightarrow{na} Dom(attr_i^n) \quad (4)$$

gdzie:

$Dom(attr_j^n)$ – oznacza zbiór wartości, jakie przyjmuje atrybut $attr_j^n$.

W oparciu o listę wymiarów i ich poziomy szczegółowości można utworzyć M typów wielowymiarowych agregatów, określonych przez kombinacje poziomów agregacji poszczególnych wymiarów.

$$M = \prod_{n=1}^N L_n \quad (5)$$

Niech

$$\mathbf{C} = \{c_1, c_2, \dots, c_M\} \quad (6)$$

oznacza zbiór typów wielowymiarowych agregatów (kostek).

Każda kostka c_i jest jednoznacznie definiowana przez n -elementowy wektor opisujący poziomy agregacji poszczególnych wymiarów.

Kostka c_i może zostać wyliczona w oparciu o kostkę c_j , jeżeli dla każdego wymiaru poziom agregacji kostki jest nie mniejszy niż poziom agregacji kostki źródłowej, czyli zachodzi:

$$\bigwedge_{n=1, N} \left(attr_{a_n}^i \preceq attr_{a_n}^j \right) \quad (7)$$

Możliwość wyliczenia zawartości agregatu w oparciu o dane przechowywane w innych agregatach można przedstawić w postaci grafu G , w którym zbiór

wierzchołków \mathbf{C} odpowiadających kostkom został rozszerzony o dodatkowy wierzchołek reprezentujący tabelę bazową (\mathbf{C}^0):

$$\mathbf{G} = \langle \mathbf{C}^0, \prec \rangle, \quad (8)$$

Relacja $c_i \prec c_j$ oznacza, że kostka $c_i \in \mathbf{C}$ może zostać wyliczona w oparciu o inną kostkę $c_j \in \mathbf{C}$, czyli zachodzi warunek (7).

Wyznaczenie schematu agregatów polega na znalezieniu podzbioru kostek \mathbf{C} , który zostanie „zmaterializowany”, to jest zostaną wyliczone wartości miar w tych agregatach i odświeżane każdorazowo po załadowaniu hurtowni nowymi danymi.

W dalszej części artykułu przyjmujemy, że zmaterializowany podzbiór agregatów jest opisany binarnymi zmiennymi decyzyjnymi:

$$x_i \in \{0, 1\}, i = \overline{1, M} \quad (9)$$

określającymi, czy kostka c_i zostanie wskazana do wyliczenia.

Graf \mathbf{G} jest wykorzystywany do znalezienia takiego planu wyliczania danych w agregatach, który przy zadanej wartości zmiennej x będzie prowadził do najkrótszego czasu trwania tego procesu.

3. Odświeżanie danych

W chwili, kiedy do tabeli bazowej wstawiane są rekordy, niezbędne jest ponowne wyliczenie zawartości kostek danych ze względu na dezaktualizację podsumowań znajdujących się w komórkach agregatów. Zmiany w tabeli bazowej pociągają za sobą konieczność aktualizacji danych w każdym z wyliczonych agregatów. Jeżeli przyjmiemy, że detaliczna hurtownia danych jest ładowana w trybie dziennym, wówczas w tym samym cyklu należy wyliczać dane w agregatach.

Dane w kostce mogą być aktualizowane w oparciu o dane z tabeli bazowej bądź innej kostki pozostającej w odpowiedniej relacji \prec . Jako koszt aktualizacji danych przyjmujemy czas wymagany do wyliczenia i zapisania zawartości zmienionych komórek agregatu. Koszt związany z aktualizacją kostki może być opisany jako funkcja na krawędziach grafu \mathbf{G} pokazującego zależności obliczeniowe pomiędzy kostkami.

W zależności od użytych algorytmów aktualizacji danych w agregatach i zakresu zrównoleglenia operacji, czas odświeżenia tego samego schematu agregatów może się różnić w odmiennych systemach zarządzania wielowymiarową bazą danych.

4. Zapytania użytkowników

Typy zapytań klasy SPJG kierowane do hurtowni danych należą do skończonego zbioru, wyznaczonego przez wymiary analizy oraz warunki selekcji.

Oznaczmy przez \mathbf{Q} zbiór numerów typów zapytań:

$$\mathbf{Q} = \{1, 2, \dots, M\} \quad (10)$$

Klasy równoważności zapytań, odpowiadające typom zapytania, zostały wyróżnione w oparciu o poziom agregacji wymiarów występujący w treści zapytania.

Każdemu typowi zapytania odpowiada dokładnie jeden typ agregatu (jest to ta kostka, w której poziomy agregacji wymiarów są identyczne jak w zapytaniu).

Jeżeli kierowane do hurtowni danych zapytanie zawiera warunki selekcji, określone na dziedzinie atrybutu, który nie występuje w klauzuli grupującej zapytania, wówczas według przyjętego modelu należy dla zadanego wymiaru wskazać poziom szczegółowości tegoż atrybutu. Odpowiedź na tego typu zapytanie może zostać udzielona wyłącznie w oparciu o agregat, w którym wymiar występuje w stopniu szczegółowości nie niższym, niż odpowiadający zadanemu atrybutowi.

Zapytania kierowane przez użytkowników do hurtowni danych mogą być modelowane jako strumień zdarzeń oraz odpowiadający mu dyskretny w stanach proces stochastyczny. Wartością $Z(t_i)$ procesu jest numer typu zapytania skierowanego do hurtowni danych w chwili t_i .

Zmienne losowe $Z(t_i), Z(t_{i+1}), \dots, Z(t_{i+k})$ są niezależnymi zmiennymi losowymi o tym samym, znanym rozkładzie prawdopodobieństwa. Możemy przyjąć, że numer typu zapytania kierowanego przez użytkownika do hurtowni w dowolnej chwili jest realizacją tej samej zmiennej losowej Z .

Rozkład zmiennej losowej Z wyznaczają wartości prawdopodobieństw wystąpienia zapytania i -tego:

$$w_i = P\{Q = i\} \quad (11)$$

Dysponując rozkładem zmiennej losowej Z można wyznaczyć zależność oczekiwanego czasu odpowiedzi $T(x)$ od podzbioru zmaterializowanych agregatów. Zakładamy przy tym, że znamy czas odpowiedzi $\tau_i(x)$ na i -te zapytanie dla schematu agregatów zdefiniowanego przez x ; przy czym:

$$T(x) = \sum w_i \cdot \tau_i(x) \quad (12)$$

Mając dany zbiór numerów typów zapytań Q oraz rozszerzony o tabelę bazową zbiór agregatów C^0 , możliwość wyznaczenia odpowiedzi na zapytanie można przedstawić w postaci dwudzielnego grafu:

$$H = \langle Q \cup C^0, E \rangle \quad (13)$$

Graf ten stanowi podstawę do wyznaczenia takiego planu wykonania zapytania przez moduł przepisywania zapytań, aby czas odpowiedzi był najmniejszy z możliwych, przy określonym zbiorze zmaterializowanych agregatów.

5. Kryteria i ograniczenia zadania optymalizacji

Oczywistym kryterium optymalizacji wydaje się być wartość oczekiwana czasu odpowiedzi na zapytanie użytkownika. Taka funkcja celu pojawia się w najstarszych chronologicznie publikacjach [4][5] i większość proponowanych algorytmów optymalizacji schematu agregatów, przyjmując jako ograniczenie przestrzeń dyskową, minimalizuje właśnie tę wielkość.

W pozycji [6] zaproponowane zostało użycie jako kryterium liczby typów zapytań, na które odpowiedź może zostać wyznaczona przy wykorzystaniu schematu agregatów.

Próbę uwzględnienia więcej niż jednego kryterium można odnaleźć w [7], gdzie minimalizowana jest suma składników: oczekiwanego czasu odpowiedzi na zapytanie oraz oczekiwanego czasu wymaganego do odświeżenia zawartości agregatów.

Głównym ograniczeniem związanym z utrzymywaniem schematu agregatów jest czas niedostępności struktur danych dla użytkowników powodowany odświeżaniem danych. W przypadku, gdy hurtownia jest ładowana wsadowo w cyklu dziennym, czas odświeżenia danych $u(x)$ nie może przekroczyć wydzielonego okna ładowania danych do hurtowni.

Przy założeniu, że dane do hurtowni spływają na bieżąco (bez opóźnień pomiędzy chwilą zaistnienia zdarzenia a datą pojawienia się informacji o zdarzeniu w systemie źródłowym), czas aktualizacji struktur danych nie zależy od okresu, w jakim dane przechowywane są w schemacie agregatów, lecz wyłącznie od listy typów zmaterializowanych agregatów.

Ograniczenie związane z przestrzenią dyskową, choć mogłoby się wydawać mało istotne w kontekście spadających cen nośników

pamięci, nie może zostać pominięte – zapotrzebowanie na ten rodzaj pamięci rośnie wykładniczo w funkcji wymiarów analizy. Dodatkowo, jest to dobry wskaźnik kosztów związanych z utrzymaniem dodatkowej warstwy danych, obejmujących koszt archiwizacji danych, koszt pracy administratorów związanej z nadzorem nad przebiegiem procesów ładowania danych oraz monitorowaniem parametrów środowiska, czy też wreszcie koszt serwisu sprzętowego.

Przestrzeń dyskowa $D(x,t)$, wymagana do materializacji schematu agregatów, zależy od maksymalnej liczby wymiarów analizy, która w dopuszczalnym uproszczeniu, w danym obszarze tematycznym, można być uznana za niezmienną, oraz długości okresu, za jaki dane są przechowywane w schemacie. Jeżeli przyjąć, że w agregatach przechowywane są kompletne dane historyczne, wymagana przestrzeń dyskowa będzie rosła liniowo w funkcji czasu. W dalszej części artykułu, przez wielkość przestrzeni dyskowej $d(x)$ będziemy oznaczali wartość wymaganą do przechowywania danych z pojedynczego interwału czasu:

$$D(x,t) = t \cdot \sigma \cdot d(x) \quad (14)$$

W artykule przedstawione zostało uogólnione podejście, w którym problem wyboru optymalnego schematu agregatów został opisany jako wielokryterialne zadanie optymalizacji.

6.

7. Wielokryterialne zadanie optymalizacji

Jednym z etapów projektowania hurtowni danych jest wskazanie, jakie agregaty zostaną wyliczone – jest to decyzja, którą podejmuje projektant hurtowni danych.

Jako kryteria przyjmujemy dwuwymiarowy wektor:

$$\langle T(x), d(x) \rangle \in R^+ \times R^+ \quad (15)$$

Ogólną postać zadania optymalizacji schematu agregatów opisano poniżej.

Dane:

1. H – graf wykonalności zapytań,
2. G – graf zależności obliczeniowych pomiędzy typami agregatów,
3. $d(x)$ – zależność opisująca wymaganą przestrzeń dyskową od typów materializowanych agregatów,
4. $T(x)$ – oczekiwany czas odpowiedzi na zapytanie przy założeniu, że zostanie ono wykonane w sposób optymalny,

tn. gwarantujący najkrótszy możliwy czas odpowiedzi, przy zadanym zbiorze wyliczonych agregatów opisanym wartością zmiennej x . Plan wykonania takiego zapytania wyznaczony jest w oparciu o graf **G**,

5. $u(x)$ – czas aktualizacji danych w kostce przy założeniu, że aktualizacja będzie wykonana w sposób optymalny, to jest gwarantujący najkrótszy możliwy czas odświeżenia danych w zbiorze agregatów opisanych zmienną x . Plan odświeżania zawartości danych w schemacie agregatów wyznaczany jest w oparciu o graf **H**,
6. U – limit czasu dla aktualizacji danych w schemacie agregatów wynikający z dostępnego okna czasowego ładowania hurtowni danych.

Szukamy $x^* = \langle x_1^*, \dots, x_M^* \rangle$ minimalizującego następujące wielkości:

$$\begin{aligned} T(x) &\rightarrow \min \\ d(x) &\rightarrow \min \end{aligned} \quad (16)$$

przy ograniczeniu:

$$u(x) < U \quad (17)$$

8. Rozwiązanie zadania

Na rys. 2. przedstawiono zbiór rozwiązań dopuszczalnych w przestrzeni kryteriów, uzyskany poprzez rozwiązanie ciągu zadań optymalizacji jednokryterialnej [8].

Wartości kryteriów zostały standaryzowane zakresowo, przy czym maksymalna wartość oczekiwanego czasu T_{max} odpowiedzi odpowiada sytuacji, w której odpowiedzi na wszystkie zapytania wyznaczane są w oparciu o tabele bazy:

$$y_1 = \frac{T(x)}{T_{max}} \quad (18)$$

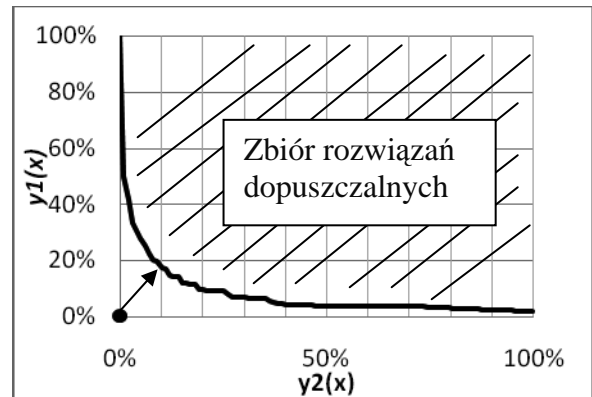
Wartość drugiego kryterium została standaryzowana analogicznie:

$$y_2 = \frac{d(x)}{d_{max}} \quad (19)$$

gdzie d_{max} oznacza maksymalną przestrzeń dyskową wymaganą do zmaterializowania wszystkich agregatów.

Na tym samym rysunku zaznaczony został dodatkowo zbiór rozwiązań niezdominowanych (optymalnych w sensie Pareto) oraz punkt

idealny mający interpretację następującą: zarówno czas odpowiedzi jak i zajmowany obszar dyskowy powinny mieć wartość jak najbliższą zeru.



Rys. 2. Zbiór rozwiązań Pareto-optymalnych

Przedstawione powyżej zadanie nie posiada rozwiązania dominującego. Satisfakcjonujący punkt można odnaleźć wyszukując punkt najbliższy punktowi idealnemu o współrzędnych $\langle 0,0 \rangle$.

Dla wyszukania takiego punktu należy odpowiednio przeskalować wielkości kryteriów, na przykład w sposób zaprezentowany na wykresie. W oryginalnej postaci, wielkości kryteriów odłożone na osiach mają różne wymiary i mierzone są w innych, nieporównywalnych jednostkach.

9. Podsumowanie

Wielokryterialne sformułowanie zadanie optymalizacji pozwala na uwzględnienie preferencji użytkownika w zakresie wagi kryteriów. Uwzględnienie więcej niż jednego kryterium w zadaniu optymalizacji umożliwia modelowanie preferencji użytkownika np. w postaci wag przypisywanych poszczególnym kryteriom.

Należy wspomnieć o innych, niż agregaty, podejściach do problemu zwiększania efektywności systemu hurtowni danych. Prekalkulacja wyników zapytań w postaci wielowymiarowych agregatów nie jest jedynym kierunkiem poszukiwań w zakresie zmniejszanie czasu odpowiedzi systemów. W systemach Business Intelligence „in-memory” [9] zakłada się, że wszystkie dane niezbędne do wyliczenia odpowiedzi na zapytanie można zmieścić w pamięci operacyjnej dzięki efektywnym algorytmom kompresji i dzięki temu ograniczyć liczbę operacji IO. W systemach ROLAP, w których nie jest budowana warstwa pośrednia,

kładzie się nacisk na przygotowanie właściwych indeksów dla relacyjnych zbiorów danych.

10. Bibliografia

- [1] M. Jarke, M. Lenzerini., Y. Vassiliou, P. Vassiliadis, *Fundamentals of Data Warehouses*, Springer-Verlag, Berlin, 2003.
- [2] *MetaObjectFacility(MOF) Specification*, <http://www.omg.org/technology/documents/formal/mof.htm/>.
- [3] *Common Warehouse Metamodel Specification*, <http://www.omg.org/spec/CWM/1.1/PDF/>
- [4] V. Harinarayan, A. Rajaraman, J.D. Ullman, *Implementing Data Cubes Efficiently*, ACM SIGMOD, 205-216, 1996.
- [5] H. Gupta, S. Mumick, *Selection of views to materialize under a maintenance cost constraint*, International. Conference on Database Theory (ICDT), 453-470, 1999.
- [6] E. Hung, D.W. Cheung, B. Kao, Y.L. Liang, *An Optimization Problem Design in Data Cube System Design* Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Lecture Notes in Artificial Intelligence 1805, 74-85, Springer-Verlag, 2000.
- [7] H. Gupta, *Selection of views to materialize in a data warehouse*. International. Conference. on Database Theory (ICDT), 98-112, 1997.
- [8] M. Mazurek, *Optymalizacja schematu wielowymiarowych agregatów w hurtowni danych*, WAT, Warszawa, 2008, rozprawa doktorska.
- [9] <http://www.qlikview.com>.
- [10] J. Kusiak, A. Danielewska-Tulecka, P. Oprocha, *Optymalizacja*, Wydawnictwo Naukowe PWN, Warszawa, 2009.

Symulator rozprzestrzeniania się złośliwego oprogramowania w sieciach komputerowych

A. NAJGEBAUER, R. KASPRZYK
e-mail: anajgebauer@wat.edu.pl

Instytut Systemów Informatycznych
Wydział Cybernetyki WAT
ul. S. Kaliskiego 2, 00-908 Warszawa

Wyraźny wzrost zainteresowania systemami dających się modelować z wykorzystaniem teorii grafów i sieci jest spowodowany rosnącym znaczeniem rzeczywistych sieci wielkiej skali. Badania niezawodności i odporności tych systemów na przypadkowe, jak i celowe ataki oraz trudne do przewidzenia awarie mają oczywiste znaczenie praktyczne. W artykule przedstawiono koncepcję modelowania i symulacji zagrożeń dla sieci teleinformatycznych. Zaprezentowano aplikację umożliwiającą symulację rozprzestrzeniania się złośliwego oprogramowania, badanie struktury i prognozowanie możliwych kierunków ewolucji sieci teleinformatycznych, optymalizację sposobów wykorzystania zasobów czy w końcu formułowanie możliwych procedur postępowania w sytuacjach kryzysowych np. przypadkowe awarie pewnych węzłów lub celowe ataki terrorystyczne.

Słowa kluczowe: sieci złożone, miary centralności, strategie szczepień

1. Wprowadzenie

W otaczającej rzeczywistości sieci rozumiane jako zbiory węzłów i gałęzi reprezentujących relacje pomiędzy węzłami są wszechobecne. Większość sieci rzeczywistych najczęściej powstających spontanicznie (ale nie tylko) ze względu na liczbę występujących w nich węzłów i powiązań pomiędzy nimi nazywane są sieciami złożonymi (ang. *Complex Network*), rozległymi lub sieciami wielkiej skali. Od lat 50-tych minionego stulecia, sieci złożone, bez widocznej zasady organizacyjnej, były opisywane przy wykorzystaniu grafów losowych (ang. *random graphs*) [1]. Powstanie pod koniec XX wieku ogólnodostępnych baz danych gromadzących informacje o topologiach sieci rzeczywistych ujawniło jednak, że mają one szereg specyficznych cech, które nie były do tej pory uwzględniane. Okazało się więc, że choć „otaczają nas sieci”, które od wielu lat są badane i opisywane to nadal nie znamy ich topologii, a tym bardziej nie rozumiemy zasad rządzących ich dynamiką, czy ewolucją. Analizy prowadzone na rzeczywistych sieciach wykazały istnienie właściwości, które nie dają się modelować za pomocą grafów losowych. W szczególności mówi się o takich cechach sieci złożonych jak względnie niewielka liczba krawędzi (graf rzadki), relatywnie mała średnica, zaskakująco mała średnia odległość pomiędzy

węzłami (hipoteza „*six degree of separation*”), wysoki stopień klasteryzacji, czy w końcu potężowy rozkład stopnia wierzchołka [9]. Linie elektryczne, połączenia komunikacyjne, sieci internetowe, WWW, sieci społeczne, ale też połączenia neuronów w mózgu, czy cykle pokarmowe (kto kogo zjada w ekosystemie) należą do kategorii sieci złożonych. Wydaje się, że istnieje specjalny rodzaj architektury grafu specyficzny dla większości naturalnych sieci. Badania tej architektury są w toku, ale już widać, że taki model opisuje zadziwiająco wiele, często pozornie niezwiązanych ze sobą, aspektów rzeczywistości.

Pod koniec XX wieku wielu badaczy zainteresowało się szczególnie siecią Internet ze względu na jej zadziwiająco szybki wzrost dostępność i istotność dla społeczeństwa. Zadziwiająca była dla nich topologia Internetu, którego przecież rozwój nie został poddany żadnym regulacjom czy planom architektonicznym. Interesująca okazuje się również ewolucja tej globalnej sieci wymiany danych. Rozwój Internetu można określić jako spontaniczny i niekoordynowany. Niektórzy badacze odnajdują wręcz podobieństwa Internetu do żywego organizmu, a jego rozwój określają jako organiczny [12].

2. Generatory sieci

Kluczową rolę w symulowaniu rozprzestrzeniania się złośliwego oprogramowania (ang. *malwares*) odgrywają dobre modele i generatory sieci komputerowych. Przez dobre rozumiane są tu generatory, które kreują sieci złożone, a więc posiadające wcześniej wspomniane cechy właściwe dla sieci rzeczywistych. W literaturze wyróżnia się dwa kluczowe modele zaproponowane pod koniec XX wieku, które znane są jako sieci małego świata (ang. *Small World*) oraz sieci bezskalne (ang. *Scale Free*) oraz liczne ich modyfikacje, rozszerzenia i uogólnienia.

Pojęcie sieci typu *Small World* wyłania się w naturalny sposób jako bardzo realistyczny, niejako pośredni przypadek pomiędzy dwoma skrajnymi sieciami: siecią regularną (ang. *Regular Network*), zwaną niekiedy pierścieniową (ang. *Ring Network*) oraz siecią losową (ang. *Random Network*). Zaczynając rozważania od sieci regularnej, okazuje się że każdy jej węzeł jest powiązany z tą samą liczbą węzłów „sąsiadów”. Powstaje pytanie, czy tego typu, idealnie regularne sieci są zjawiskiem częstym. Przecież nawet w krytalografii zdarzają się „defekty”, zaburzające monotonną strukturę powiązań. Praktyka dowodzi, że sieci regularne są oczywiście idealizacją rzeczywistości, co znacznie redukuje ich przydatność w tworzeniu modeli sieci rzeczywiście występujących w naturze i cywilizacji. Z drugiej strony sieć losowa wykazuje całkowity brak regularności w liczbie powiązań między węzłami, jak również brak jakiegokolwiek strukturalnego uporządkowania tych powiązań. W obu przypadkach nazwy sieci dość wiernie oddają ich naturę – pełna regularność i całkowita przypadkowość. Watts i Strogatz [2] zauważyli, że dokonując pewnego zabiegu na sieci regularnej można otrzymać modele sieci, które spotykane są w rzeczywistych systemach. Nie są one bowiem ani doskonale regularne, ani zupełnie losowe, a można je budować poprzez zastosowanie tzw. „przepinania” (ang. *rewiring*) niektórych gałęzi sieci. Modele sieci *Small World* przyczyniły się do rozpowszechnienia hipotezy „*six degree of seperation*”, według której każde dwie osoby na świecie są połączone drogą zawierającą sześć społecznych powiązań. Pokazuje się również dlaczego ta najkrótsza droga jest spontanicznie odnajdywana[3].

Sieci *Scale Free* [4] można scharakteryzować, używając żargonu z obszaru sieci komputerowych jako sieci powiązań, w której kluczową rolę pełnią huby. Tego typu węzły określane są jako „*super-spreaders*”. Barabasi i Albert dostrzegli, że sieci takie „rosną przez dodawanie” kolejnych węzłów według określonej hierarchii. Prawdopodobieństwo wystąpienia połączenia pomiędzy nowym węzłem, a każdym węzłem należącym do sieci bardzo silnie zależy od posiadanej już przez te węzły liczby krawędzi k i wynosi $P(k) \sim k^{-\gamma}$. Wykładnik potęgi, a więc γ , zależy od rodzaju rozpatrywanej sieci. Jako następstwo powyższego, sieci typu *Scale Free* „rosną przez dodawanie” kolejnych węzłów według określonej hierarchii. Tak więc, najwięcej nowych węzłów łączy się z węzłami, które posiadają już najwięcej sąsiadów (połączeń z innymi węzłami). Cecha ta jest określana jako „dołączenia preferencyjne” (ang. *preferential attachment*). Istnieje wiele modyfikacji algorytmów generujących sieci *Scale Free* [9]. Modyfikacje polegają głównie na zmianie możliwości ewolucji sieci, i tak np. w kolejnych krokach ewolucji możemy mieć do czynienia nie tylko z dodaniem nowego węzła wraz z nowymi krawędziami, ale również z dodaniem jedynie krawędzi do już istniejących węzłów, czy z przepięciem wybranych krawędzi.

3. Miary centralności

Analiza dynamiki rozprzestrzeniania się złośliwego oprogramowania w sieciach komputerowych i próba przeciwdziałania epidemii nie jest możliwa bez zdefiniowania miar centralności (istotności) węzłów [8]. Wprowadzone miary istotności węzłów, dzięki istnieniu wyraźnej interpretacji fizycznej, pozwalają na ciekawą analizę topologii sieci teleinformatycznych tj. precyzyjne klasyfikowanie istniejących/ projektowanych sieci, ze względu na ich niezawodność, z uwzględnieniem kontekstu stanowiącego cel istnienia/projektowania sieci. Wierzchołki centralne sieci są szczególnie interesujące ponieważ pełnią one kluczowe role stanowiąc swego rodzaju katalizatory epidemii tj. węzły, które wpływają znacząco na jej dynamikę tzw. „*super-spreaders*”. Miary centralności ułatwiają udzielenie odpowiedzi na pytanie „kto (co) jest najważniejszy(e) w analizowanej sieci?”. Okazuje się, że nie ma jednoznacznej odpowiedzi. Wszystko zależy od przyjętej semantyki słowa „istotny”, co obrazuje pięć wprowadzonych dalej miar centralności.

Stopień wierzchołka

Najprostszą miarą centralności wierzchołka, którą podpowiada intuicja, jest stopień wierzchołka. Według tej miar wierzchołek jest tym istotniejszy dla sieci im ma więcej bezpośrednich połączeń z pozostałymi wierzchołkami w sieci.

$$center_i^{Degree} = \frac{k_i}{n-1} \quad (1)$$

gdzie k_i oznacza stopień i -tego wężła, n – liczba wierzchołków sieci.

Promień wierzchołka

Jeśli wierzchołek jest tym ważniejszy im jego odległość do najdalszego wierzchołka jest najmniejsza, wówczas należy zastosować miarę istotności wyliczaną w oparciu o promień wierzchołka. Algorytm odnalezienia wężła centralnego składa się z dwóch kroków. W pierwszej pętli dla każdego wierzchołka w sieci wyliczamy jego promień, a następnie w drugim kroku szukamy wężła dla którego promień jest najmniejszy.

$$center_i^{Radius} = \frac{1}{\max_j d_{ij}} \quad (2)$$

gdzie d_{ij} najkrótsza droga wyrażana minimalną liczbą krawędzi, które łączą węzeł v_i z v_j .

Średnia odległość wierzchołka

Bardzo często zależy nam na tym, aby dokonany wybór w większości możliwych scenariuszy (średnio rzecz biorąc) był najlepszy. Przy takim założeniu „dobrą” miarą oceny istotności wierzchołka będzie wyliczenie jego średniej odległości do wszystkich pozostałych węzłów w sieci. Wierzchołek, który „średnio” jest najbliższy wszystkim wierzchołkom w sieci jest wówczas najistotniejszy.

$$center_i^{Closeness} = \left[\frac{\sum_{j=1}^n d_{ij}}{n-1} \right]^{-1} = \frac{n-1}{\sum_{j=1}^n d_{ij}} \quad (3)$$

Obciążenie wierzchołka

Ciekawym sposobem oceny istotności wierzchołka w sieci jest wyznaczenie tzw. obciążenia wężła, które można zdefiniować jako procent najkrótszych dróg pomiędzy dowolną parą wierzchołków przechodzących przez rozpatrywany węzeł. Jeśli przez $p_{ik}(i)$ oznaczymy liczbę najkrótszych dróg pomiędzy węzłem v_i i v_k przechodzących przez węzeł v_i oraz przez p_{ik} liczbę wszystkich dróg pomiędzy węzłem v_i i v_k , wówczas:

$$center_i^{Efficiency} = \frac{2 \sum_{k \neq i} p_{ik}(i)}{(n-2)(n-1)} \quad (4)$$

Obciążenie wierzchołka jest niezwykle istotne dla odpowiedzi na pytanie: jak trudne

i czasochłonne może być zadanie polegająca na maksymalizacji rozpojenia sieci w celu zminimalizowania możliwego obszaru objętego epidemią. Usunięcie wierzchołka o największej wartości obciążenia powoduje znaczące zwiększenie średniej odległości pomiędzy węzłami, a tym samym największe utrudnienia komunikacji w sieci.

Istotność sąsiedztwa wierzchołka

O ile stopień wierzchołka za wartość istotności przyjmuje liczbę jego najbliższych sąsiadów, to w rzeczywistości oczywistym jest, że nie wszyscy sąsiedzi wierzchołka powinni zwiększać jego istotność o tą samą wartość. Jeśli wierzchołek ma znaczną liczbę połączeń, ale z wierzchołkami, które nie są istotne dla sieci to nie powinien on być uznany za tak ważny jak wierzchołek, który posiada choćby jedno połączenia, ale za to z najważniejszym węzłem w sieci. Tak więc istotność wężła v_i oznaczona jako e_i powinna zależeć od istotności wierzchołków z jakimi jest on połączony i możemy ją wyliczyć następująco:

$$e_i = \frac{1}{\lambda} \sum_{j=1}^n A_{ij} e_j \Rightarrow \vec{e} = \frac{1}{\lambda} A \vec{e} \quad (5)$$

Wartość λ (wartość własną macierzy) wyliczamy z $\det(A - \lambda I) = 0$. Wektor \vec{e} jest wektorem własnym macierzy A dla największej wartości λ .

4. Efektywność komunikacji

Do oceny stopnia uszkodzenia sieci w przypadku awarii będącej następstwem ataku bądź zdarzenia losowego, jak i efektywności strategii dystrybucji oprogramowania anty-wirusowego użyto miary zwanej *global connection efficiency* (GCE) [7]. Przy założeniu, że wydajność połączenia pomiędzy węzłem v_i i v_j jest odwrotnie proporcjonalna do najkrótszej drogi pomiędzy tymi węzłami mamy:

$$connection_{ij}^{efficiency} = \frac{1}{d_{ij}} \quad (6)$$

Interesująca jest efektywność komunikacji nie tyle pomiędzy wybranymi węzłami, ale w całej sieci przed i po ataku, jak i efektywności strategii szczepień węzłów sieci teleinformatycznej wg zadanego kryterium. Należy wyliczyć więc miarę *global connection efficiency* zdefiniowaną jako średnia wartość miara *connection efficiency* pomiędzy każdą parą węzłów.

$$GCE = \frac{2}{n(n-1)} \sum_{i < j} \frac{1}{d_{ij}} \quad (7)$$

Warto w tym miejscu podkreślić, że o ile spadek wartości GCE w przypadku awarii pewnego wężła świadczy o zmniejszeniu

efektywności komunikacji w sieci, to w przypadku porównywania efektywności strategii szczepień oznacza zmniejszenie możliwości rozprzestrzeniania się danego wirusa bez wpływu na funkcjonalność sieci jako całości. Tak więc miara GCE została wykorzystana w celu identyfikacji węzłów, które należy szczególnie chronić przed atakiem, jak również węzłów które muszą podlegać pełnej ochronie antywirusowej, gdyż mogą stać się szczególnymi katalizatorami epidemii ze względu na ich wpływ na globalną efektywność komunikacji w sieci teleinformatycznej.

5. Modelowanie rozprzestrzeniania się złośliwego oprogramowania

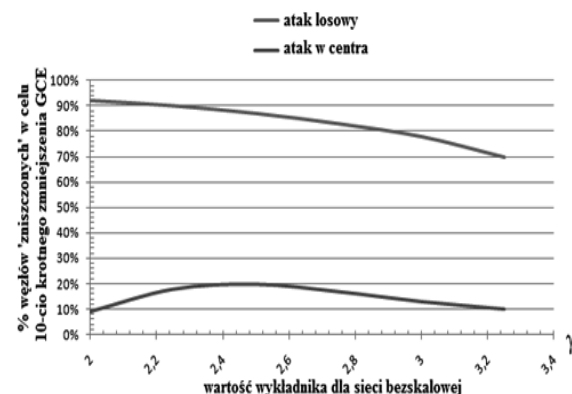
Internet, który zrodził się z chęci stworzenia solidnego systemu komunikacji mogącego przetrwać obcą agresję jest coraz częściej postrzegany jako infrastruktura niezwykle wrażliwa, od której zależy bezpieczeństwo państwa. Aby przeprowadzić skuteczny atak na tę infrastrukturę nie trzeba mobilizować sił militarnych. Człowiek wyposażony w standardowe technologie komputerowe, posiadający odpowiednią wiedzę, może przeprowadzić cyberatak o skutkach wręcz katastrofalnych dla współczesnego systemu polityczno-gospodarczego.

Jednym z zagrożeń jest przejęcie kontroli nad siecią teleinformatyczną, co wydaje się być możliwe, ale dość skomplikowane. Znacznie prostsze i prawdopodobnie równie niekorzystne byłoby przeprowadzenie mniej wyrafinowanego ataku polegającego na sparaliżowaniu sieci. Kamieniem milowym w badaniach nad odpornością Internetu na różnego rodzaju ataki było odkrycie bezskalowej topologii tej sieci. Wprowadzone miary centralności oraz współczynnik GCE pozwoliły na porównanie odporności sieci bezskalowej na awarie spowodowane losowym jak i celowym atakiem. Przeprowadzona analiza dowodzi, że charakterystyczna długość ścieżki praktycznie wcale nie zmienia się, gdy 10% jej węzłów zostanie losowo zniszczonych, co więcej dopiero losowe zaatakowanie 80% węzłów powoduje 10-krotny spadek wartości GCE. Ponieważ większość węzłów w sieci bezskalowej ma niewielką liczbę połączeń, losowe eliminowanie węzłów nie ma większego wpływu na komunikację pomiędzy pozostałymi węzłami.

Internet ma więc strukturę, jaka jest potrzebna, by zapewnić niezawodną komunikację, nawet wówczas, gdy nie wszystkie węzły działają. Lokalna niewydolność nie wynika jedynie

z awarii węzłów, gdyż często węzeł może stać się tymczasowo beużyteczny, bo jest zablokowany przez masę przechodzącej przez niego informacji. W sieci bezskalowej można łatwo znaleźć alternatywną drogę, nawet jeśli wiele węzłów jest jednocześnie zablokowanych. Co ciekawe w dowolnym momencie około 3 procent routerów w sieci Internet jest zablokowana, a sieć mimo to doskonale działa. Bezskalowa struktura Internetu gwarantuje, że próby eliminowania losowo wybranych węzłów będą prawdopodobnie nadaremne, ponieważ sieć może wytrzymać sporą liczbę takich ataków bez większej utraty zdolności komunikacji.

Mimo swej solidności i efektywności, Internet ma pewną wadę, która może okazać się fatalna w skutkach. Czy cyberterrorysta będzie bowiem próbował tak mało wyrafinowanego ataku? O ile w sieci losowej, żaden z węzłów nie ma szczególnego charakteru. W sieci bezskalowej pewne węzły są zdecydowanie ważniejsze (węzły centralne), co powoduje, że po ich usunięciu sieć ulega dezintegracji. Okazuje się, że już usunięcie 15% węzłów centralnych powoduje 10-krotny spadek wartości GCE, a tym samym brak możliwości efektywnej komunikacji w sieci. Słabością sieci bezskalowych jest również duże prawdopodobieństwo pojawienia się kaskady awarii, gdyż usterka w jednym z centrów, przenosi obciążenie na inne węzły mogące powodować ich przeciążenie. Zatem, aby sparaliżować Internet należy zidentyfikować stosunkowo niewielką liczbę węzłów, a następnie uczynić je celem ataku. Wniosek jest prosty, najbardziej istotne węzły należy wyjątkowo dobrze chronić tj. zbudować dobre mechanizmy ochronne i/lub należałoby uczynić je „ukrytymi”.



Rys. 1. Wyniki symulacji ataku na sieć bezskalową oceniane z wykorzystaniem GCE.

Rozmyślne niszczenie węzłów, jak i ich przypadkowe awarie to nie jedyne zagrożenie dla sieciowych infrastruktur krytycznych.

Wirusy komputerowe, które rzadko uszkadzają fizycznie węzły sieci, wykorzystują raczej topologię sieci, by nią „zawładnąć” dla własnych celów. Wirusy komputerowe zwykle rozprzestrzeniają się poprzez zainfekowane pakiety danych przekazywane z jednego komputera do drugiego, podobnie zresztą jak wirusy biologiczne przenoszą się pomiędzy ludźmi jako cząsteczki we wdychanym powietrzu lub przez płyny ustrojowe. Gdy pomiędzy węzłami nastąpi tak rozumiane połączenie to mogą się one od siebie zarazić.

Jak się okazuje sieci teleinformatyczne, nie poddają się uodpornieniu przeciwko złośliwemu oprogramowaniu, w wyniku strategii losowego szczepienia węzłów. Bardzo szybko jednak stają się odporne w wyniku szczepienia węzłów centralnych tj. posiadających najwyższe wartości opisanych miar centralności. Obserwacja ta okazuje się niezwykle przydatna przy planowaniu szczepień mających zapobiec rozprzestrzenianiu się epidemii złośliwego oprogramowania.

Standardowym podejściem w epidemiologii jest uproszczające założenie, że choroba zakaźna rozprzestrzenia się w populacji modelowanej jako graf losowy lub regularny. Model ten przewiduje pewien próg epidemii. Choroba szerzy się w populacji, bezustannie zarażając pewien stały odsetek ludzi, jeśli tempo jej rozprzestrzeniania się jest większe od pewnej wartości progowej, a w innym wypadku szybko zanika. Wydaje się, że niektóre epidemie faktycznie zachowują się w ten sposób. Proóg epidemii ma kluczowe znaczenie, jeśli bowiem pewien odsetek populacji będzie zaszczepiony, to tempo szerzenia się epidemii pozostanie poniżej wartości progowej, a w związku z tym choroba nie przetrwa w epidemii.

Internet nie jest jednak grafem losowym. Stefan Bornholdt i jego współpracownicy z Uniwersytetu w Kiel pokazali, że połączenia e-mailowe również tworzą sieć bezskalową, co sugeruje, że „sieć znajomych” zdefiniowana na sieci elektronicznej ma taki sam charakter jak Internet [10]. Fizycy Romualdo Pastor-Satorras w Barcelonie i Alessandro Vespignani w Trieście odkryli, że fakt bezskalowej topologii sieci po której wędrują e-maile całkowicie zmienia sposób rozprzestrzeniania się wirusów komputerowych. Korzystając z symulacji komputerowej w celu zbadania, jak zachowuje się choroba zakaźna w sieci bezskalowej, odkryli, że nie występuje tam próg szerzenia się epidemii [6]. Niezależnie jak wolne jest tempo rozprzestrzeniania się wirusa, może on przeniknąć cały system, zarażając pewien

odsetek węzłów. Ponieważ zarażone węzły mogą zostać „wyleczone” dzięki programom komputerowym, wirus w końcu zanika. Proces ten zachodzi jednak bardzo powoli. Oprogramowanie zwalczające dany wirus z reguły staje się dostępny już po kilku dniach lub tygodniach od infekcji, a jednak wirusy mogą przetrwać w sieci nawet przez wiele lat.

Czy sytuacja jest tak fatalna? Jak wiadomo, powodem takiego zachowania sieci bezskalowych jest nieproporcjonalny wpływ pewnych węzłów na rozprzestrzenianie się chorób. Jeśli przerwie się połączenia do tych kluczowych węzłów to cała sieć szybko rozpadnie się. Pastor-Satorras i Vespignani pokazali, że nakierowanie szczepień ochronnych na jednostki prowadzące bogate życie seksualne zdecydowanie obniża wrażliwość sieci na epidemie chorób przenoszonych drogą płciową [11]. Na tej samej zasadzie wybuch epidemii wirusa komputerowego może zostać skutecznie powstrzymany poprzez „zaszczepienie” zaledwie 15% wierzchołków wybranych ze względu na liczbę połączeń.

Niezwykle interesujące jest to, że Internet rozrósł się bez żadnego planu w tę, jak się wydaje, najbardziej solidną z możliwych struktur sieci. Gdyby istniała organizacja pod dyktando której należałoby dokonywać kolejnych przyłączeń węzłów to zapewne powstała struktura nie byłaby tak solidna. Czasami więc najlepiej jest pozwolić by system sam się organizował. Pozostaje jednak pytanie, dlaczego Internet ma taką właśnie strukturę?

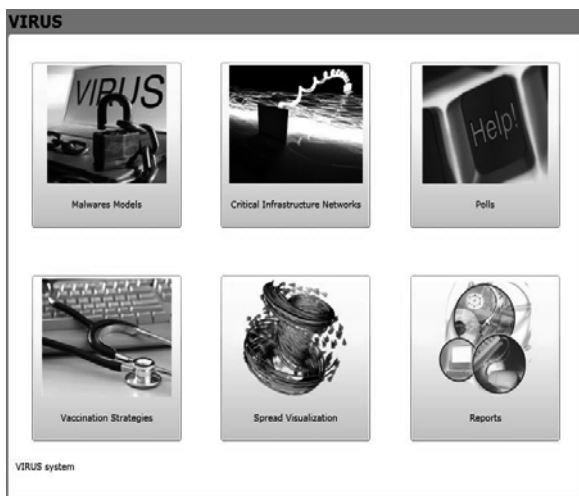
6. Funkcjonalność aplikacji do badania odporności sieci teleinformatycznych

Symulator został zaimplementowany na platformie .NET 3.5 z SP1. Pierwotnie projektowany był jako aplikacja desktopowa. Ostatnie pozytywne doświadczenia z aplikacjami webowymi doprowadziły do całkowitej modyfikacji architektury. Obecnie aplikacja działa zgodnie z koncepcją SOA (ang. *Service Oriented Architecture*). Interfejs użytkownika natomiast został stworzony w oparciu o rozwiązanie AJAX (ang. *Asynchronous JavaScript and XML*) z wykorzystaniem *Microsoft Silverlight 3.0*.

Funkcjonalność aplikacji można sprowadzić do kilku kluczowych obszarów:

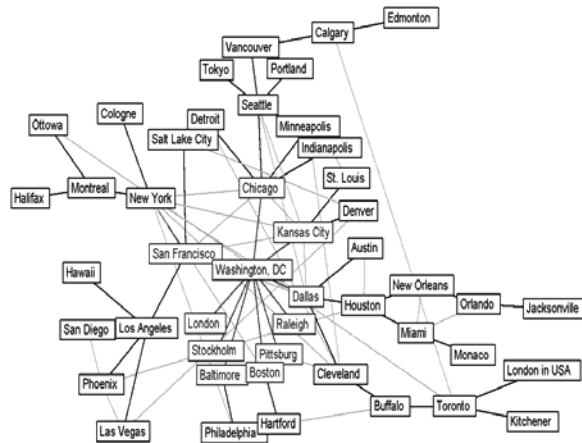
- a) Modelowanie złośliwego oprogramowania (wirusy, robaki, trojany, itp.) mającego szkodliwe, przestępcze lub złośliwe

- działanie w stosunku do użytkownika komputera – moduł *Malwares Models*;
- b) Generowanie modeli sieci o topologiach rzeczywistych sieci teleinformatycznych z wykorzystaniem zaimplementowanych generatorów sieci złożonych – moduł *Critical Infrastructure Networks*;
 - c) Konstrukcja kwestionariuszy wykorzystywanych do zbierania danych dotyczących topologii wirtualnej sieci kontaktów pomiędzy użytkownikami komputerów – moduł *Polls*;
 - d) Identyfikacja węzłów sieci teleinformatycznej istotnych z punktu widzenia odporności sieci na awarie, jak i strategii dystrybucji oprogramowania antywirusowego z wykorzystaniem zaimplementowanych miar centralności. Weryfikacja skuteczności proponowanych strategii z wykorzystaniem proponowanej miary GCE – moduł *Vaccination Strategies*;
 - e) Symulacja i wizualizacja możliwych scenariuszy rozprzestrzeniania się złośliwego oprogramowania modelowanego z wykorzystaniem modelu stworzonego w punkcie a) na sieci wygenerowanej w punkcie b) lub wirtualnej sieci kontaktów zidentyfikowanej za pomocą kwestionariuszy z punktu c). Scenariusze tworzone są z pomocą modułu d) – moduł *Spread Visualization*;
 - f) Ocena oczekiwanych skutków awarii węzłów i/lub epidemii złośliwego oprogramowania oraz szacowanie środków (programów antywirusowych) niezbędnych do przeciwdziałania lub zahamowania epidemii – moduł *Reports*.



Rys. 2. Główne okno aplikacji do badania odporności sieci teleinformatycznych

Jednym z bardziej efektownych elementów prezentowanej aplikacji jest moduł e), który prócz symulacji służy do interaktywnej wizualizacji powiązań pomiędzy węzłami sieci. Analiza tego typu pozwala na badanie topologii sieci komputerowej. Na tej podstawie ułatwione staje się zadanie identyfikacji kluczowych węzłów, analiza możliwych sposobów komunikowania się w sieci i wiele innych. Diagramy powiązań mogą służyć również do przedstawiania i analizy dużych zbiorów danych, dotyczących np. rozmów telefonicznych, transakcji na kontach bankowych czy ruchu internetowego. W takich przypadkach diagramy posłużą do łatwej identyfikacji wspólnych elementów, grup elementów ściśle powiązanych ze sobą czy też powiązań pośrednich. Waga wizualizacji powiązań wydaje się być, w tym przypadku, nie do przecenienia.



Rys. 3. Zobrazowanie sieci UUNET z wykorzystaniem zbudowanej aplikacji.

7. Podsumowanie

W artykule opisano wpływ topologii sieci Internet na jej wrażliwość na awarie oraz sposób rozprzestrzeniania się złośliwego oprogramowania. Zaproponowano model symulacyjny pozwalający symulować rozprzestrzenianie się złośliwego oprogramowania w sieciach o różnych topologiach, w tym sieciach złożonych. Otrzymane wyniki pozwalają odpowiedzieć na pytanie, czy i jeśli tak to kiedy, występują różnice w skutkach celowego ataku na sieć w porównaniu ze skutkami uszkodzeń spowodowanych zjawiskami losowymi. Pokazano dlaczego standardowa wiedza epidemiologiczna jest niewystarczająca do walki z wirusami komputerowymi. Przeprowadzone symulacje

dowodzą również konieczności weryfikacji istniejącego wyobrażenia o szerzeniu się epidemii w sieciach społecznych.

W chwili obecnej symulator jest prototypem systemu nad rozwojem którego wciąż trwają intensywnej prace. Jednak już na obecnym etapie symulator wykazuje bogatą funkcjonalność i dowodzi możliwości praktycznego jego wykorzystania.

8. Literatura:

- [1] P. Erdős, A. Rényi: *On random graphs*, Publ. Math Debrecen 6 , 290-297, 1959.
- [2] Watts Duncan J., Strogatz Steven H.: *Collective dynamics of „small-world” networks*, Nature, 393:440-442, 1998.
- [3] Jon M. Kleinberg: *Navigation in small world*, Nature 406, 845 (24 August 2000)
- [4] Barabási Albert László, Albert Réka: *Emergency of Scaling in Random Networks*, Science, 286:509-512, 1999.
- [5] Brandes U., Kenis P., and Raab J.: *Explanation Through Network Visualization*, Methodology, Vol. 2(1):16-23, 2006.
- [6] Romualdo Pastor-Satorras & Alessandro Vespignani: *Epidemic Spreading in Scale-Free Networks*, PRL Vol. 86, Number 14 p. 3200 (2 April 2001).
- [7] Paolo Crucitti, Vito Latora, Massimo Marchiori, Andrea Rapisarda: *Error and attack tolerance of complex networks*, Physica A 340, 388-394, 2004.
- [8] Stefan Wuchty, Peter F. Stadler: *Centers of complex networks*, Journal of Theoretical Biology 222, 45-53, 2003.
- [9] Barabási Albert László, Albert Réka, *Topology of Evolving Networks: Local Events and Universality*, PRL Vol. 85, Number 24, p.5234 (11 December 2000).
- [10] H. Ebel, L.I. Mielsch, S. Bornholdt, *Scale-free topology of email networks* Phys. Rev. E 66, 035103(R) (2002).
- [11] Romualdo Pastor-Satorras & Alessandro Vespignani: *Immunization of complex networks*, PRL Vol. 65, 036104 (2002).
- [12] Philip Ball: *Critical mass: How one thing leads to another*, New York, Farrar, Straus & Giroux, 2004.

Metoda analizy ruchu drogowego na podstawie zachowania użytkowników sieci telefonii komórkowej

G.SABAK

e-mail:gsabak@wat.edu.pl

Wydział Cybernetyki WAT
ul. S. Kaliskiego 2, 00-908 Warszawa

Artykuł ten poświęcony jest możliwości wykorzystania informacji o przemieszczaniu się użytkowników sieci telefonii komórkowej do analizy sytuacji na drogach objętych zasięgiem tej sieci.

W pracy przedstawiony został formalny model danych lokalizacyjnych dostępnych w sieci, model drogi jako linii łamanej oraz model pokrycia drogi zasięgiem telefonii komórkowej. Przy użyciu wymienionych modeli sformułowany został problem uzyskiwania *informacji o ruchu drogowym*.

Artykuł zawiera propozycję metody wyznaczania informacji o ruchu na podstawie strumienia danych lokalizacyjnych. W celu wstępnej weryfikacji zaproponowanej metody, wykonane zostały obliczenia dla trasy Kraków-Zakopane, dla danych zebranych z sieci jednego z operatorów telefonii komórkowej. Wyliczona informacja drogowa została porównana z, uzyskaną na podstawie przejazdu testowego, informacją referencyjną.

Słowa kluczowe: lokalizacja CellId, transport, zatłoczenia

1. Wprowadzenie

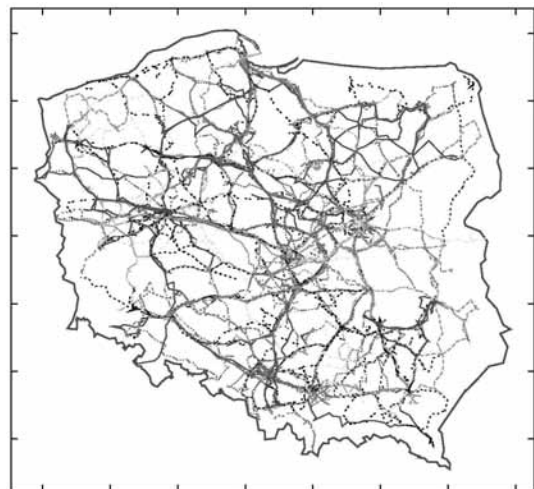
Transport i telekomunikacja mobilna, ze względu na ich powszechne wykorzystanie, mają dziś olbrzymie znaczenie zarówno dla gospodarek państw i regionów, jak i dla poszczególnych obywateli. Można wymienić następujące cechy wspólne dla obu tych dziedzin:

- zajmują się zapewnianiem dostępności dóbr (transport) i informacji (telekomunikacja) w określonym miejscu i czasie
- w obu występuje konieczność zapewnienia oczekiwanej efektywności procesów przy jednoczesnym istnieniu ograniczeń na koszty ich działania
- problemy występujące w obu dziedzinach są podobne. Mianowicie są to m.in.: znajdowanie efektywnych dróg transportu, wykrywanie i unikanie blokad oraz zatłoczeń.

Te podobieństwa oraz fakt, że bardzo wiele osób jednocześnie korzysta z systemu transportu i z sieci telefonii komórkowej, skłaniają do przeanalizowania możliwości wykorzystania informacji o zmieniającej się lokalizacji użytkowników komórek do wnioskowania o stanie systemu drogowego.

Na rys. 1. przedstawione zostały wybrane ślady stacji mobilnych należących do sieci telefonii komórkowej jednego z operatorów w Polsce. Uwzględnione zostało 200 stacji

mobilnych, które w dniu 6.08.2009 r. przemieściły się na największe odległości.



Rys. 1. Ślady przemieszczeń wybranych stacji mobilnych

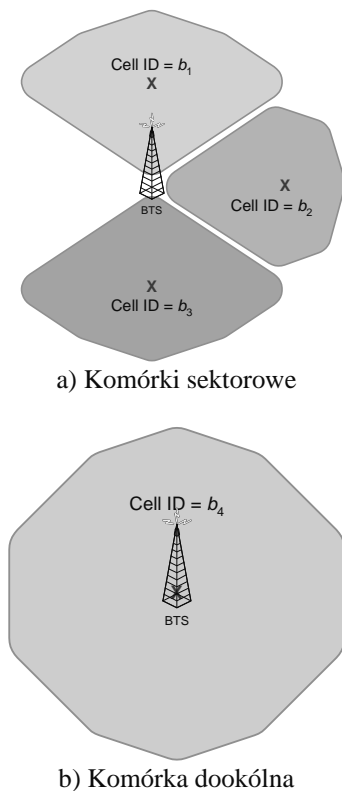
Na rysunku widoczne jest, że ślady te pokrywają się z przebiegiem głównych szlaków transportowych w naszym kraju. Potwierdza to przypuszczenie, że istnieje grupa użytkowników telefonów komórkowych, którzy w tym samym czasie korzystają z transportu samochodowego.

1.1. Metoda lokalizacji w sieci GSM na podstawie CellId

Jedną z najczęściej wykorzystywanych przez operatorów sieci komórkowych metod

lokalizacji użytkowników jest wykorzystanie informacji o identyfikatorze komórki (CellId), która w danej chwili obsługuje stację mobilną. Obsługa ta polega na realizacji komunikacji (zarówno przez kanał głosowy jak i sygnalizacyjny) pomiędzy stacją mobilną, a częścią stałą sieci telefonii komórkowej. W tej metodzie każdemu z identyfikatorów komórki przypisywane są współrzędne geograficzne wskazujące miejsce, które przyjmuje się za lokalizację stacji mobilnej.

Podstawowym ograniczeniem tej metody jest mała dokładność lokalizacji, która wynosi ok. 300-500 metrów dla obszaru gęstej zabudowy miejskiej i 35-50 km dla obszarów pozamiejskich [1]. Dodatkowo, ponieważ na ogół obszary obsługiwane przez komórki pokrywają się, często występuje sytuacja, w której dla pewnego położenia stacji mobilnej, lokalizacja uzyskiwana metodą CellId jest skokowo różna w sąsiednich chwilach.



Rys. 2. Różne typy komórek

Na rys. 2. przedstawione zostały schematycznie dwa rodzaje komórek występujące w sieci: sektorowa i dookólna. Typowo, dla komórki dookólnej lokalizacja zwracana przez metodę CellId (zaznaczona na rysunku znakiem X) jest tożsama z położeniem wieży stacji bazowej (BTS). Dla komórek sektorowych zwracana lokalizacja jest z reguły

bliska środka obszaru, jaki obsługuje dana komórka.

Inne metody lokalizacji użytkowników w sieci GSM korzystają z: parametrów czasowych komunikacji stacji mobilnej z siecią, informacji o sile sygnału radiowego, danych z trzech i więcej stacji bazowych, informacji z systemu GPS [2, 3]. Zapewniają one większą dokładność lokalizacji, jednak bardzo wysokie koszty związane z ich implementacją powodują, że niewielu operatorów decyduje się na ich realizację i uruchomienie.

1.2. Wstępna analiza problemu

Trudności związane z praktyczną realizacją pomysłu wnioskowania o stanie systemu transportowego na podstawie przemieszczania się użytkowników sieci komórkowej są następujące:

- informacja o lokalizacji jest bardzo niedokładna, z praktycznego punktu widzenia użyteczne informacje o sytuacji na drodze wymagają dokładności przynajmniej o rząd wielkości większej
- brak jest informacji podanej wprost o tym, które stacje mobilne są użytkownikami systemu transportowego
- w praktyce, dostęp do informacji o wszystkich użytkownikach jest niemożliwy ze względu na koszty, które musiałby ponieść operator. Koszty te wynikają ze znacznego rozproszenia zarządzania obsługą użytkowników i tego, że są to podsystemy specjalizowane, zamknięte, bez interfejsów pozwalających na łatwy dostęp do przetwarzanych przez nie danych.

Celem opisywanej pracy badawczej było zaproponowanie i weryfikacja metody, przy pomocy której, pomimo powyżej przedstawionych trudności, operator sieci komórkowej będzie mógł uzyskać informację o systemie transportowym na obszarze, który jest obsługiwany przez jego sieć. Informacja ta powinna być na tyle dokładna i aktualna, by na jej podstawie opłacalne było przygotowanie usług, które mogłyby być następnie zaoferowane klientom instytucjonalnym i korporacyjnym.

Na podstawie uzyskiwanych informacji, w ogólnym przypadku, możliwe byłoby rozwiązywanie następujących problemów praktycznych:

- analiza strumienia ruchu na drodze w zależności od dnia tygodnia, pory dnia itp.
- wykrywanie blokad i zagęszczeń ruchu

- znajdowanie fragmentów dróg o niezadawalającej przepustowości.

Te potencjalne zastosowania są dodatkową motywacją do zajęcia się opisanym powyżej zagadnieniem.

2. Model danych i definicja problemu

2.1. Definicje i oznaczenia

Obiektem mobilnym nazywany będzie obiekt świata rzeczywistego poruszający się po powierzchni Ziemi. Przykładem obiektu mobilnego będącego w zakresie zainteresowania może być np. pojazd, telefon komórkowy, moduł GSM lub odbiornik GPS. *Komórką* nazywamy obszar zasięgu sygnału radiowego emitowanego przez jedną ze stacji przekaźnikowych. *Stacją mobilną* nazywany będzie zestaw składający się z urządzenia technicznego użytkownika (wraz z oprogramowaniem) wykorzystywany do komunikacji w sieci telefonii komórkowej.

Wprowadźmy następujące oznaczenia:

- N – zbiór liczb naturalnych
- R – zbiór liczb rzeczywistych,
- $A \subset R^2$ – obszar, na którym „działa” sieć telefonii komórkowej
- $t \in T \subset R$ – czas, znacznik czasu
- $p \in A$ – punkt na obszarze (płaszczyźnie), lokalizacja
- $d(p_1, p_2)$ – odległość (euklidesowa) pomiędzy punktami $p_1 = \langle x_1, y_1 \rangle$ i $p_2 = \langle x_2, y_2 \rangle$, obliczana wg wzoru:

$$d(p_1, p_2) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

- $O = \{1, 2, \dots, N_o\}$ – zbiór identyfikatorów obiektów mobilnych o
- $\lambda : O \times T \mapsto A$ – funkcja, która przypisuje obiektowi jego lokalizację w chwili t
- $\lambda^{\text{GPS}} : O \times T \mapsto A$ – funkcja, która przypisuje obiektowi jego lokalizację w chwili t wyznaczoną przez system GPS.

2.2. Model sieci telefonii komórkowej

Wprowadźmy następujące oznaczenia:

- B – zbiór identyfikatorów b komórek wchodzących w skład sieci komórkowej. Niech 0 oznacza „brak komórki”

$$B = \{0, 1, 2, \dots, N_B\}$$

- M – zbiór identyfikatorów stacji mobilnych m

$$M = \{1, 2, \dots, N_M\}$$

- $\theta : M \times T \mapsto B$ – funkcja przypisująca stacji mobilnej identyfikator komórki, przez którą, w chwili t , jest ona obsługiwana. Jeśli $\theta(m, t) = 0$, to oznacza to, że w chwili t stacja mobilna m nie jest widoczna w sieci komórkowej

- $\beta : B \mapsto A$ – funkcja przypisująca każdej z komórek lokalizację interpretowaną jako miejsce, w którym znajduje się stacja mobilna obsługiwana przez tę komórkę. Punkt $p^b = \beta(b)$ nazywaliśmy będziemy *środkiem komórki b* .

Metodą lokalizacji w sieci komórkowej będziemy nazywać funkcję $M \times T \mapsto A$, która stacji mobilnej m przypisuje dla danej chwili t pewne położenie geograficzne.

Metodą lokalizacji opartą o CellId będziemy nazywać funkcję określoną wzorem:

$$\lambda^{\text{CellId}}(b, t) = \beta(\theta(b, t)) \quad (2)$$

Niech piątka $\Lambda = \langle A, B, M, \theta, \beta \rangle$ będzie modelem sieci telefonii komórkowej działającej na obszarze A , ze zdefiniowaną metodą lokalizacji w oparciu o CellId.

2.3. Model danych lokalizacyjnych

Zdarzeniem nazywana będzie para $e = \langle t, P \rangle$, gdzie t jest znacznikiem czasu określającym chwilę, w której miało miejsce zdarzenie, a P to lista parametrów definiujących typ zdarzenia. *Strumieniem zdarzeń* nazywamy ciąg zdarzeń

$$E = \{e_1, e_2, \dots\}$$

uporządkowanych niemalejąco względem znacznika czasu t .

Funkcję $\phi : e \rightarrow \{0, 1\}$ będziemy określać mianem *filtru strumienia zdarzeń*. Wartość 1 przypisana zdarzeniu będzie oznaczała, że zostało ono przepuszczone przez filtr. Wartość 0 oznaczała będzie, że filtr odrzucił zdarzenie e . Operację, która z ciągu E tworzy nowy ciąg przez usunięcie tych zdarzeń, dla których $\phi(e) = 0$, nazywaliśmy będziemy *operacją filtrowania*.

Zdarzeniem lokalizacyjnym l nazywane będzie zdarzenie, w którym $P = \langle o, p \rangle$. Zdarzenie lokalizacyjne zawiera informację o fakcie znajdowania się obiektu o w chwili t w lokalizacji p .

Zdarzeniem lokalizacyjnym CellId l^b nazywamy zdarzenie, w którym $P = \langle m, b \rangle$, gdzie $b = \theta(t, m)$. *Strumieniem zdarzeń*

lokalizacyjnych $CellId$ nazywamy ciąg zdarzeń lokalizacyjnych

$$L^b = \{l_1^b, l_2^b, \dots\}$$

uporządkowanych niemalejąco względem znacznika czasu t .

2.4. Model drogi

Niech modelem drogi będzie linia łamana otwarta, zbudowana z odcinków, których końcami są kolejne punkty ciągu

$$\hat{Q} = \{\hat{q}_1, \hat{q}_2, \dots, \hat{q}_{N_Q}\}$$

Zbiór punktów Q należących do linii łamanej oznaczany będzie przez $Q \subset A$

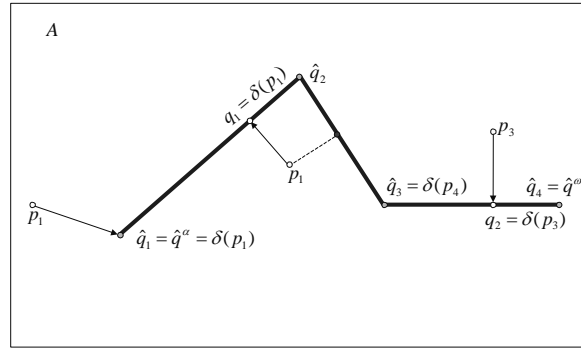
Punkty będące wyrazami ciągu \hat{Q} będą nazywane *punktami wierzchołkowymi drogi*, a odcinki $\hat{q}_k \hat{q}_{k+1}$ *segmentami drogi*. Punkt $\hat{q}^\alpha = \hat{q}_1$ nazywany będzie *punktem początkowym drogi*, a punkt $\hat{q}^\omega = \hat{q}_{N_Q}$ jej *punktem końcowym*.

Pikietażem nazywać będziemy funkcję $\pi: Q \mapsto R_{\geq 0}$ przypisującą wszystkim punktom należącym do drogi ich odległość wzdłuż linii łamanej licząc od punktu początkowego drogi. Można zauważyć, że $\pi(\hat{q}^\alpha) = 0$. Wartość $\pi(\hat{q}^\omega)$ jest długością całkowitą drogi, a różnica $\pi(\hat{q}_{k+1}) - \pi(\hat{q}_k)$ jest długością segmentu $\hat{q}_k \hat{q}_{k+1}$.

Dla przedziału $\langle 0, \pi(\hat{q}^\omega) \rangle$ określona jest

funkcja $\pi^{-1}(x): R \mapsto Q$ odwrotna do funkcji pikietażu, której wartościami są punkty na drodze leżące w odległości x od początku drogi.

Dla drogi Q , zdefiniujemy funkcję $\delta(p)$ jako funkcję $A \mapsto Q$ przypisującą każdemu z punktów należących do A punkt drogi najbliższy w sensie odległości euklidesowej. Jeśli istnieje wiele takich punktów, to wybierany jest ten o najmniejszym pikietażu. Operację znajdowania $q = \delta(p)$ będziemy nazywać przyciąganiem punktu do drogi. Zasady przyciągania punktów do modelu zilustrowane zostały na rys. 3.

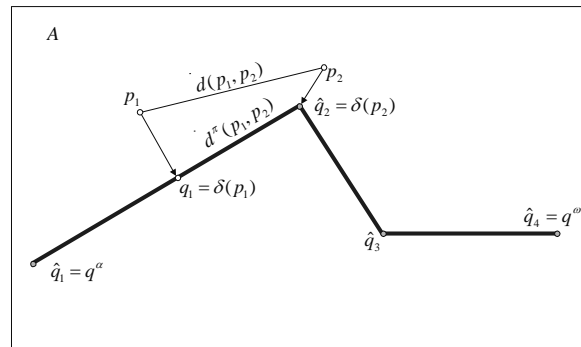


Rys. 3. Przyciąganie punktów do drogi

Odległością pikietażową punktów $p_1, p_2 \in A$ będziemy nazywali wielkość wyliczoną wg następującej definicji:

$$d^\pi(p_1, p_2) = |\pi(\delta(p_1)) - \pi(\delta(p_2))|. \quad (3)$$

Przykład pokazujący różnicę pomiędzy obliczaniem odległości euklidesowej, a odległości pikietażowej przedstawiony został na rys. 4. Zauważmy, że odległość pikietażowa dwóch punktów może być zarówno większa jak i mniejsza od ich odległości euklidesowej.



Rys. 4. Odległość euklidesowa, a odległość pikietażowa

2.5. Model pokrycia drogi zasięgiem sieci telefonii komórkowej

Zdefiniujemy *funkcję zasięgu* $\gamma: A \times B \mapsto \{0,1\}$ jako funkcję przyporządkowującą wartość 1 tym punktom obszaru A , które są w zasięgu komórki, a wartość 0 tym punktom, które są poza jej zasięgiem. Poprzez bycie punktu p w zasięgu komórki rozumiemy fakt, że stacja mobilna znajdująca się w punkcie p może być w pełni obsługiwana przez tę komórkę.

Funkcją pokrycia drogi przez komórkę będziemy nazywali taką funkcję

$$\gamma^\pi: \langle 0, \pi(\hat{q}^\omega) \rangle \times B \mapsto \{0,1\},$$

która przyporządkowuje wartość 1 tym wartościom pikietażu, które odpowiadają

punktom drogi będącym w zasięgu komórki. W pozostałych punktach funkcja przyjmuje wartość 0.

Pokryciem drogi przez komórkę b nazywamy zbiór $c_b^Q \subseteq \langle 0, \pi(\hat{q}^\omega) \rangle$ zdefiniowany jak poniżej:

$$c_b^Q = \left\{ x : x \in \langle 0, \pi(\hat{q}^\omega) \rangle \wedge \gamma^\pi(x, b) = 1 \right\} \quad (4)$$

Jeżeli $c_b^Q = \emptyset$, to będziemy mówili, że droga Q nie jest w zasięgu komórki o identyfikatorze b .

Pokryciem drogi Q przez sieć telefonii komórkowej będziemy nazywali zbiór zasięgów wszystkich komórek, w których zasięgu jest droga Q :

$$C^Q = \{c_1^Q, c_2^Q, \dots, c_{N_C}^Q\}$$

Dla danej drogi Q i pokrycia C^Q definiujemy filtr zdarzeń ϕ^c wzorem:

$$\phi^c(l^b) = \begin{cases} 1 & \text{gdy } l^b = \langle t, m, b \rangle \wedge c_b^Q \neq \emptyset, \\ 0 & \text{w przeciwnym wypadku.} \end{cases} \quad (5)$$

Początkiem pokrycia drogi przez komórkę b będziemy nazywali wartość $c_b^<$ określoną przez poniższy wzór:

$$c_b^< = \min c_b^Q \quad (6)$$

Końcem pokrycia drogi przez komórkę b będziemy nazywali wartość $c_b^>$ określoną jak poniżej:

$$c_b^> = \max c_b^Q \quad (7)$$

Z powyższych definicji wynika, że $c_b^< \leq c_b^>$.

Przez C^q oznaczany będzie zbiór tych komórek, w których zasięgu znajduje się punkt $q \in Q$.

Długością zasięgu komórki b będziemy nazywali wielkość $\bar{c}_b \in R_{\geq 0}$ obliczaną zgodnie z poniższym wzorem:

$$\bar{c}_b = \int_0^{\pi(\hat{q}^\omega)} \gamma^\pi(x, b) dx \quad (8)$$

Środkiem ciężkości zasięgu komórki b , będziemy nazywali liczbę

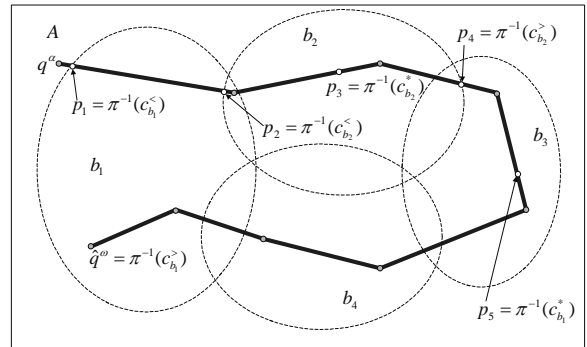
$$c_b^* = \frac{1}{\bar{c}_b} \int_0^{\pi(\hat{q}^\omega)} x \gamma^\pi(x, b) dx \quad (9)$$

Przy założeniu, że prawdopodobieństwo znajdowania się stacji mobilnej w każdym miejscu drogi jest takie samo, punkt $\pi^{-1}(c_b^*)$ może być uznawany za oczekiwane miejsce

znajdowania się stacji mobilnej będącej w zasięgu komórki b . Dodatkowo zauważmy, że nie musi być spełnione $c_b^* \in c_b^Q$, czyli że środek ciężkości zasięgu nie musi być wartością należącą do pokrycia drogi przez komórkę.

Można zauważyć, że jeśli pokrycie drogi przez komórkę jest zbiorem spójnym (przedziałem), to $c_b^* = \frac{c_b^> + c_b^<}{2}$.

Na rys. 5. pokazane zostało przykładowe pokrycie drogi zasięgiem sieci telefonii komórkowej.



Rys. 5. Pokrycie drogi przez sieć komórkową

Zaznaczone zostały punkty odpowiadające początkom, końcom i środkom ciężkości zasięgów komórek b_1 i b_2 . W przypadku komórki b_1 , środek ciężkości jej zasięgu (punkt p_5 na Rys. 5) nie należy do jej zasięgu.

2.5.1. Odległość dwóch komórek

Odległością dwóch komórek $b_i, b_j \in B$ będziemy nazywali funkcję $d: B \times B \mapsto R_{\geq 0}$ spełniającą warunki:

$$\begin{aligned} d(b_i, b_i) &= 0 \wedge \\ d(b_i, b_j) &= d(b_j, b_i) \wedge \\ d(b_i, b_k) &\leq d(b_i, b_j) + d(b_j, b_k) \end{aligned} \quad (10)$$

Na potrzeby dalszych rozważań wprowadzmy następujące definicje:

- odległością zwykłą dwóch komórek będziemy nazywać funkcję: $\bar{d}(b_i, b_j) = d(\lambda^{\text{CellId}}(b_i), \lambda^{\text{CellId}}(b_j))$ (11) czyli odległość euklidesową środków komórek.

- odległością pikietażową dwóch komórek będziemy nazywać funkcję $\hat{d}(b_i, b_j) = d^\pi(\lambda^{\text{CellId}}(b_i), \lambda^{\text{CellId}}(b_j))$ (12)

Inaczej mówiąc jest to odległość pikietażowa ich środków

- odległością *pokryciową* dwóch komórek o identyfikatorach b_i i b_j będziemy nazywać funkcję określoną następującym wzorem:

$$\tilde{d}(b_i, b_j) = |c_{b_j}^* - c_{b_i}^*| \quad (13)$$

Można zauważyć, że ponieważ wielkości te są definiowane jako odległość euklidesowa dwóch punktów na płaszczyźnie (dla odległości zwykłej) bądź jako odległość dwóch punktów na linii (odległości pikietażowa i pokryciowa), spełniają warunki (10).

2.5.2. Charakterystyki pokrycia drogi siecią komórkową

Zdefiniujemy *współczynnik krotności pokrycia* jako wielkość określoną poniższym wzorem:

$$c_x = \frac{\sum_{i=1}^{N_B} \tilde{c}_i}{\pi(\hat{q}^\omega)} \quad (14)$$

Funkcją *krotności pokrycia* $\tilde{c}(q)$ będziemy nazywali funkcję $\langle 0, \pi(\hat{q}^\omega) \rangle \mapsto N$, zdefiniowaną wzorem:

$$\tilde{c}(x) = C^q, \quad (15)$$

która pikietażowi, każdego z punktów należących do drogi przypisuje liczbę komórek w których zasięgu znajduje się ten punkt.

2.6. Model informacji o stanie systemu transportowego

Informacją o ruchu na drodze będziemy nazywali ciąg skończony

$$S = \{s_1, s_2, \dots, s_{N_s}\},$$

którego elementami są trójki $s = \langle q, u^+, u^- \rangle$, gdzie

- $q \in Q$ - punkt na drodze, którego dotyczy informacja
- $u^+, u^- \in \{0, 1, 2, 3\}$ - stan systemu drogowego dla ruchu odpowiednio: w kierunku rosnącego pikietażu i kierunku malejącego pikietażu. Wartościom u^+, u^- nadajemy następującą interpretację:
 - 0 - 'unknown' - brak danych, sytuacja nieokreślona,
 - 1 - 'low' - zatłoczenie, ruch powolny,
 - 2 - 'medium' - ruch utrudniony,
 - 3 - 'high' - ruch swobodny.

W celu ocenienia jak wybrana informacja o ruchu różni się od innej informacji o ruchu

(przy założeniu, że obie dotyczą tego samego zbioru punktów) wprowadźmy następującą funkcję oceny różnicy w informacji dla ustalonego punktu drogi i dla ustalonego kierunku:

$$u_d(u^*, u) = \begin{cases} 0 & \text{gdy } u^* = 0 \\ |u - u^*| & \text{gdy } u^* \neq 0 \text{ i } u \neq 0 \\ 2 & \text{gdy } u^* \neq 0 \text{ i } u = 0 \end{cases} \quad (16)$$

W powyższej definicji informację u^* nazywamy informacją referencyjną. Pod pojęciem informacji referencyjnej rozumiana jest informacja, która jest traktowana jako „pewna”. Może ona pochodzić z wykonanych pomiarów, lub być informacją zadaną w eksperymencie symulacyjnym.

Funkcję porównującą dwie informacje o ruchu (dla ciągu punktów) będziemy oznaczali jako s_d i definiowali wg poniższego wzoru:

$$s_d(s^*, s) = \frac{\sum_{i=1}^{N_s} u_d(u_i^{*+}, u_i^+) + \sum_{i=1}^{N_s} u_d(u_i^{*-}, u_i^-)}{2N_s} \quad (17)$$

Wartością różnicy pomiędzy dwoma informacjami o ruchu jest wartość średnia różnicy informacji obliczona dla poszczególnych punktów wchodzących w skład informacji o ruchu przy uwzględnieniu obu kierunków. Funkcja s_d przyjmuje wartość 0, kiedy to uznajemy, że informacje o ruchu są identyczne, wtedy, kiedy wartości u są takie same dla wszystkich punktów, w których informacja referencyjna ma wartości inne niż 'unknown'.

2.7. Sformułowanie problemu

W oparciu o wprowadzone powyżej definicje i oznaczenia, problem może być sformułowany następująco: Dla danych Λ , \hat{Q} , C^Q i L^b oraz dla określonego przedziału czasu $\langle t_1, t_2 \rangle$ wyznaczyć *zadowalającej jakości* informację o stanie ruchu drogowego.

Na obecnym etapie prac pojęcie *zadowalającej jakości* nie jest jeszcze formalnie definiowane. Jakość rozwiązań jest oceniana poprzez subiektywną ocenę użyteczności uzyskiwanych informacji.

3. Rozwiązanie problemu

3.1. Uzupełniające definicje i oznaczenia

Zdefiniujemy funkcję $v(q)$ jako funkcję $Q \mapsto R_{\geq 0}$ przyporządkowującą punktom drogi

pewną wielkość, która jest proporcjonalna do średniej prędkości pojazdów w tym miejscu. Wielkość tą będziemy nazywali *szacowaną prędkością*.

Profilom prędkości W dla drogi Q będziemy nazywać ciąg

$$W = \{w_1, w_2, \dots, w_{N_w}\}$$

krotek $w = \langle q, v^+, n^+, v^-, n^- \rangle$, gdzie:

- q – punkt drogi, którego dotyczy informacja
- v^+ – szacowana prędkość pojazdów poruszających się w kierunku rosnącego pikietażu
- n^+ – liczba zdarzeń, dla których została obliczona wartość v^+
- v^- – szacowana prędkość pojazdów poruszających się w kierunku malejącego pikietażu
- n^- – liczba zdarzeń, dla których została obliczona wartość v^- .

3.2. Wyznaczanie profilu prędkości

Dla danych: A , Λ , i Q , profil prędkości będzie wyznaczany na podstawie L^b wg następującego ogólnego algorytmu VelProfile (t_{move}, d_{min}), który przedstawiony został w tabeli 1. Algorytm ten przyjmuje dwa następujące parametry:

- t_{move} – długość odcinka czasu w przeszłości, jaka jest analizowana
- d_{min} – minimalna wymagana odległość, jaką musiała pokonać stacja mobilna w ciągu czasu t_{move} poprzedzającego zdarzenie, by zostało ono uwzględnione w wyliczeniu szacowanej prędkości.

Tabela 1. Algorytm wyznaczania profilu prędkości

VelProfile (t_{move}, d_{min})
1. Zainicjuj W jako ciąg pusty.
2. Dla każdego zdarzenie lokalizacyjnego $l^b = \langle t, m, b \rangle$ należącego do L^b :
a) Znajdź $q' := \pi^{-1}(c_b^*)$.
b) Jeśli istnieje $w \in W$, dla którego $q = q'$ to podstaw $w' := w$. W przeciwnym wypadku do W dodaj $w' = \langle q', 0, 0, 0, 0 \rangle$.

c) Znajdź $\hat{l}^b = \langle \hat{t}, \hat{m}, \hat{b} \rangle$ takie, że $\hat{m} = m$ i $\tilde{d}(b, \hat{b})$ jest maksymalne spośród zdarzeń, dla których $t - \hat{t} \leq t_{move}$.

d) Na podstawie c_b^* i $c_{\hat{b}}^*$ wyznacz kierunek ruchu związany ze zdarzeniem l^b .

e) Jeśli $\tilde{d}(b, \hat{b}) < d_{min}$ przejdź do analizy następnego zdarzenia i wróć do punktu 2a).

f) Oblicz szacowaną prędkość związaną ze zdarzeniem wg wzoru

$$v' = \frac{\tilde{d}(b, \hat{b})}{t - \hat{t}} \quad (18)$$

g) Jeśli zdarzenie jest związane z ruchem w kierunku *rosnącego* pikietażu, to zmodyfikuj w' następująco:

$$w' := \left\langle q', \frac{n^+ v^+ + v'}{n^+ + 1}, n^+ + 1, v^-, n^- \right\rangle \quad (19)$$

h) Jeśli zdarzenie jest związane z ruchem w kierunku *malejącego* pikietażu, to zmodyfikuj w' następująco:

$$w' := \left\langle q', v^+, n^+, \frac{n^- v^- + v'}{n^- + 1}, n^- + 1 \right\rangle \quad (20)$$

3.3. Wyznaczanie informacji drogowej

Wyznaczanie S na podstawie W odbywa się zgodnie z algorytmem TraffInfo ($itval, n_{min}$), który przedstawiony został w tabeli 2. Parametrami algorytmu są:

- $itval$ – odległość od punktu q , dla której zliczane są elementy ciągu W
- n_{min} – minimalna liczba zdarzeń potrzebna żeby uznać wyliczoną informację o ruchu drogowym za istotną.

Tabela 2. Algorytm wyznaczania informacji o ruchu

TraffInfo ($itval, n_{min}$)
1. Zainicjuj S jako ciąg elementów $\langle q, 0, 0 \rangle$, dla punktów q , dla których ma zostać wyznaczona informacja o statusie.
2. Dla każdego elementu $s = \langle q, u^+, u^- \rangle$ ciągu S :
a) Wyznacz przedział pikietażu, dla którego zliczane są szacowane prędkości. Dla kierunku rosnącego pikietażu będzie to:
$\langle \max\{0, \pi(q) - itval\}, \pi(q) \rangle$,

a) dla kierunku malejącego pikietażu:

$$\langle \pi(q), \min\{\pi(q) + itval, \pi(\hat{q}^o)\} \rangle$$

b) Dla obu kierunków oblicz wartości v jako średnią ważoną szacowanych prędkości dla tych $w \in W$, które należą do,

odpowiedniego dla kierunku, przedziału pikietażu.

c) Dla obu kierunków oblicz wartości n jako sumę liczby zdarzeń dla tych $w \in W$, które należą do, odpowiedniego dla kierunku, przedziału pikietażu.

d) Dla obu kierunków oblicz informację o ruchu drogowym wg poniższego ogólnego wzoru:

$$u = \begin{cases} 0 & \text{gdy } n < n_{\min} \\ 1 & \text{gdy } n \geq n_{\min} \wedge v < 30 \\ 2 & \text{gdy } n \geq n_{\min} \wedge 30 \leq v < 60 \\ 3 & \text{gdy } n \geq n_{\min} \wedge v \geq 60 \end{cases} \quad (21)$$

Progi dla wartości v w równaniu (21) zostały wybrane arbitralnie, jako reprezentujące subiektywną ocenę warunków na drodze. Ich dobór powinien być przedmiotem dokładniejszej analizy w przyszłości.

4. Wyniki obliczeniowe

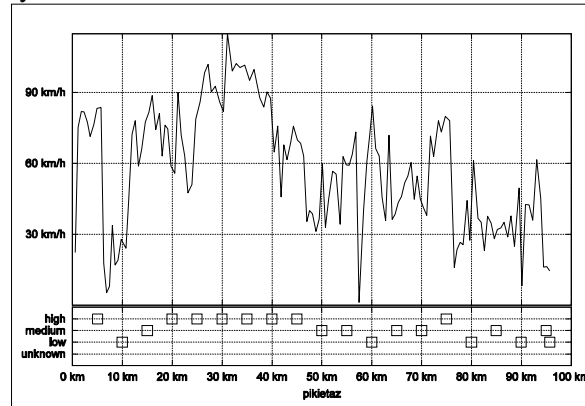
W celu weryfikacji zaproponowanej metody przeprowadzone zostały obliczenia dla trasy prowadzącej z Krakowa do Zakopanego (zwanej potocznie Zakopianką). Długość analizowanej trasy wynosi 95.71 km. Podczas obliczeń porównane zostały uzyskiwane różnice pomiędzy wyliczonymi informacjami drogowymi a informacją referencyjną, którą traktujemy jako wiarygodny opis rzeczywistej sytuacji na drodze.

W przedstawionych poniżej obliczeniach ograniczono się do analizy ruchu tylko w jednym kierunku (rosnącego pikietażu).

4.1. Referencyjna informacja drogowa

Ponieważ szczegółowe dane o rzeczywistej sytuacji na drodze nie są dostępne, do obliczeń wykorzystana została informacja drogowa przygotowana na podstawie pojedynczego przejazdu testowego na analizowanej trasie. Przejazd ten miał miejsce 6 sierpnia 2009 roku pomiędzy godziną 13-tą a 16-tą w kierunku od Krakowa do Zakopanego. Założono, że zarejestrowana prędkość pojedynczego pojazdu odpowiada szacowanej średniej prędkości pojazdów poruszających się drogą i że sytuacja

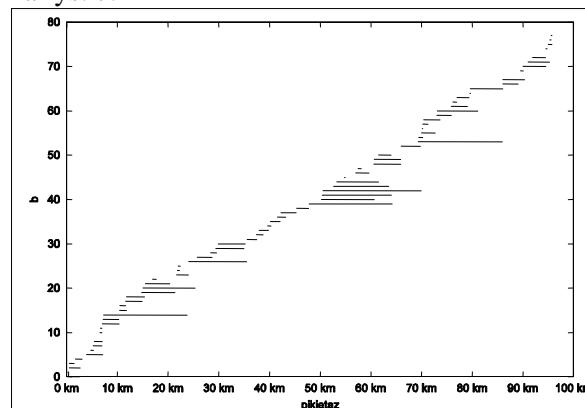
na drodze w tym czasie odpowiada typowej sytuacji na drodze w ciągu dnia w tym okresie. Na jej podstawie została wygenerowana referencyjna informacja drogowa dla zbioru punktów rozłożonych na drodze równomiernie, co 5 km. Prędkość przejazdu oraz uzyskana informacja drogowa zostały przedstawione na rys. 6.



Rys. 6. Referencyjna informacja o ruchu na trasie Kraków-Zakopane

4.2. Pokrycie trasy zasięgiem sieci komórkowej

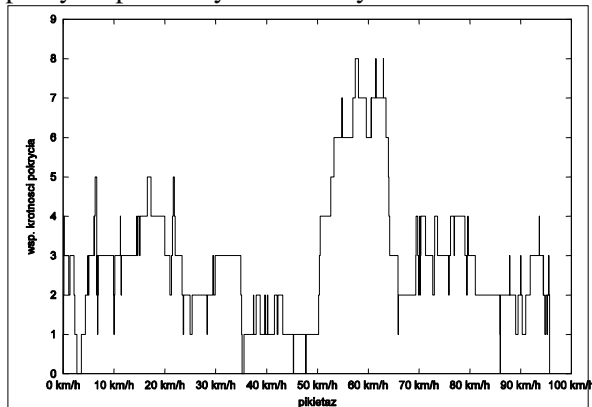
W celu określenia pokrycia analizowanej drogi zasięgiem sieci komórkowej wykonano jazdę testową, podczas której monitorowano i logowano identyfikatory komórek, które obsługiwały w danym miejscu każdą z sześciu testowych stacji mobilnych. Monitorowanie wartości CellId odbywało się co 20 sekund. Zasięgi poszczególnych komórek zostały przedstawione na rys. 7.



Rys. 7. Zasięgi i położenie komórek

Trasa Zakopianka pokrywana jest przez zbiór 78 komórek, o zasięgu od kilkuset metrów do kilkunastu kilometrów. Średnia długość zasięgu komórki to 3.60 km. Komórka o maksymalnym zasięgu pokrywa odcinek drogi o długości 19.31 km.

Przebieg wykresu funkcji krotności pokrycia pokazany został na rys. 8.



Rys. 8. Indeks pokrycia drogi siecią komórkową

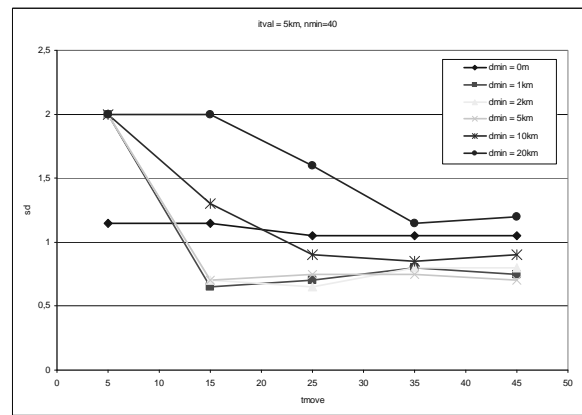
Widać na nim, że na większej części droga pokryta jest przez 2-3 komórki a maksymalna wartość funkcji krotności pokrycia wynosi 8. Obliczony dla analizowanej trasy współczynnik krotności pokrycia wynosi $c_x = 2.93$.

Jako strumień danych lokalizacyjnych wykorzystane zostały logi platformy Gateway Mobile Location Centre (GMLC) działającej w sieci jednego z operatorów komórkowych. Platforma GMLC jest odpowiedzialna za przekazywanie danych o położeniu abonentów serwerom realizującym lokalizacyjne usługi dodane. Zaletą takiego podejścia jest to, że poprzez logi, dane lokalizacyjne są dostępne w bardzo prosty sposób i niskim kosztem (znajdują się w plikach binarnych, które mogą być zgrane z platformy protokołem ftp). Wadą jest to, że informacja dotyczy jedynie niewielkiej części abonentów - tylko tych, którzy korzystali z usług dodanych oferowanych przez operatora.

Analizowane były dane z okresu od 1. do 21. sierpnia 2009 z godzin 8:00÷20:00. Przed dalszymi obliczeniami na strumieniu danych lokalizacyjnych wykonana została operacja filtrowania przy użyciu filtra ϕ^c , co spowodowało, że uwzględniane były tylko te zdarzenia, dla których b było w zbiorze komórek pokrywających Zakopiankę.

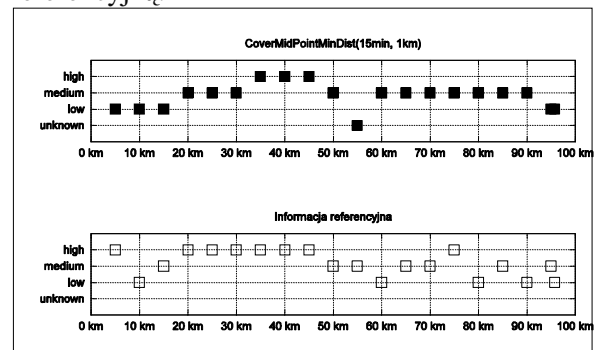
4.3. Rezultaty obliczeń

Uzyskane wartości różnicy pomiędzy informacją wyznaczoną a referencyjną, dla różnych wartości parametrów zostały przedstawione na wykresie na rys. 9.



Rys. 9. Wartości s_d dla uzyskanych informacji drogowych

Z obliczeń wynika, że informację najbardziej zbliżoną do referencyjnej uzyskano dla wartości parametrów d_{min} i t_{move} wynoszących odpowiednio 1 km i 15 minut oraz 2 km i 25 minut. Na rys. 10. zestawiona została jedna z uzyskanych informacji z informacją referencyjną.



Rys. 10. Zestawienie informacji o ruchu drogowym

Na podstawie analizy wyników można wyciągnąć wniosek, że zaproponowana metoda daje możliwość uzyskiwania informacji o stanie systemu drogowego. Jednak dla przyjętych założeń i wykorzystanego zestaw danych wejściowych nie było możliwe uzyskanie informacji równej informacji referencyjnej. Z jednej strony może to być spowodowane niedoskonałością metody, ale z drugiej tym, że informacja referencyjna mogła nie do końca odzwierciedlać standardowej sytuacji na drodze w tym okresie.

5. Podsumowanie

W artykule przedstawiona została metoda analizy danych lokalizacyjnych generowanych w elementach sieci telefonii komórkowej. Celem takiej analizy jest wnioskowanie o sytuacji na wybranej trasie, która jest pokryta zasięgiem sieci.

Przedstawione wyniki obliczeniowe pokazują na przykładzie trasy z Krakowa do Zakopanego, że przy odpowiednim doborze parametrów, możliwe jest uzyskanie informacji o ruchu, który jest bliski informacji referencyjnej, która z założenia opisuje typowy stan na drodze. Dalsze badania mogą obejmować:

- weryfikację dla jakich wartości parametrów zaproponowanej metody analizy zdarzeń lokalizacyjnych uzyskiwane są najlepsze rezultaty
- rozszerzenie badania o statystyczną analizę istotności wyników, np. potraktowanie profilu prędkości jako ciągu statystyk wyznaczających wartość oczekiwaną prędkości w danym punkcie drogi i obliczenie przedziałów ufności dla poszczególnych wartości
- weryfikację poprzez symulację jak zmienia się informacja o ruchu w zależności od napływających danych lokalizacyjnych, jeśli zmienia się realna sytuacja na drodze. Pozwoli to odpowiedzieć na pytanie, przy jakich warunkach (okres monitorowania i liczba monitorowanych stacji mobilnych) możliwe jest wykrywanie zatłoczeń w zadanym czasie
- zwiększenie ilości danych wejściowych przez dołączenie informacji lokalizacyjnej z innych elementów sieci komórkowej niż tylko platformy GMLC
- analizę, jaki wpływ na wyniki działania metody ma przyjęta procedura wyznaczania pokrycia drogi przez sieć i czy np. wykorzystanie informacji o pokryciu, która znajduje się w systemach wspomagających projektowanie sieci u operatora mogłoby poprawić jakość uzyskiwanych wyników.

ekonomicznych i prawnych. Etap 1: Charakterystyka metod służących do określania lokalizacji abonentów w sieciach GSM i UMTS., Instytut Łączności, Warszawa, 2007.

6. Bibliografia

- [1] *CGALIES final report*, Report on implementation issues related to access to location information by emergency services (E112) in the European Union, 2002.
- [2] 3G TS 23.171 3rd Generation Partnership Project; Technical Specification *Group Services and System Aspects; Functional stage 2 description of location services in UMTS*, (Release 1999).
- [3] W. Michalski, *Przegląd metod określania lokalizacji abonentów w ruchomych publicznych sieciach komórkowych GSM/UMTS z uwzględnieniem dokładności dostarczanej informacji, technicznych możliwości wdrożenia oraz czynników*

Określanie priorytetów zmiennych pewnych funkcji decyzyjnych

J. WIŚNIEWSKA

e-mail: joanna.wisniewska@wat.edu.pl

Instytut Systemów Informatycznych
Wydział Cybernetyki WAT
ul. S. Kaliskiego 2, 00-908 Warszawa

W pracy opisane są dwie metody wyznaczania priorytetów zmiennych pewnych funkcji logicznych (takich, w których zapisie nie występuje negacja). Pierwsza metoda opiera się na wykorzystaniu miary Hamminga w procesie określania priorytetów zmiennych. Ze względu na konieczność generowania i przeglądania tablicy prawdy dla badanej funkcji, złożoność czasowa tej metody jest rzędu wykładniczego. Druga proponowana metoda polega na analizowaniu zapisu funkcji w postaci minimalnej formuły sumacyjnej. Daje ona wyniki mniej precyzyjne niż pierwsza wspomniana metoda, ale nie wymaga podania tablicy prawdy dla danego zadania, dzięki czemu jest zazwyczaj mniej złożona czasowo od metody używającej miary Hamminga.

Słowa kluczowe: priorytet zmiennej, funkcja decyzyjna, odległość Hamminga

1. Charakterystyka rozważanych funkcji decyzyjnych

Podane w tym referacie metody określania priorytetów zmiennych nie mogą być z powodzeniem stosowane dla dowolnych funkcji decyzyjnych. Metody te zostały opracowane z myślą o pewnych funkcjach logicznych.

Definicja 1: Funkcją logiczną nazywamy dowolne odwzorowanie postaci:

$$f : X \rightarrow Y \quad (1)$$

gdzie, przy $B = \{0;1\}$, X jest podzbiorem B^n

$$B^n = B \times B \times \dots \times B \quad (2)$$

zaś Y jest podzbiorem B^m .

Założenie 1: W pracy rozważana jest funkcja decyzyjna $f(x)$, która ma postać funkcji logicznej n zmiennych, przy czym w zapisie $f(x)$ nie występuje negacja.

2. Określanie priorytetów zmiennych decyzyjnych za pomocą miary Hamminga

Spostrzeżenie 1: Jeżeli można zauważyć co najmniej jedną własność zmiennych x_i (dla $i = 1, n$) funkcji decyzyjnej $f(x)$, to zmienne x_i można podzielić na grupy według wartości

przyjmowanej przez obserwowaną własność. Zmienne o tej samej wartości badanej własności, w aspekcie tejsze cechy, przyjmujemy za równoważne.

Posiadając tabelę prawdy dla danego zadania można sprawdzić jaka jest relacja między wartościami przyjmowanymi przez zmienne x_i a wartością funkcji $f(x)$. Niech miarą pozwalającą ustalić wartość takiej relacji będzie odległość Hamminga.

Definicja 2: Odległość Hamminga jest to miara odmienności dwóch ciągów o takiej samej długości. Wyraża ona liczbę miejsc (pozycji), na których analizowane ciągi posiadają nie równe sobie elementy.

Oznaczenie 1: Za pomocą $H(x_i)$ będziemy określać wartość mówiącą o odległości Hamminga między ciągiem reprezentującym wartości przyjmowane przez zmienną x_i a ciągiem wartości funkcji $f(x)$.

Zgodnie ze spostrzeżeniem 1, dwie zmienne tej samej funkcji decyzyjnej $f(x)$, np. x_i i x_j , dla których zachodzi: $H(x_i) = H(x_j)$; traktujemy jako równoważne.

Przykład 1: Niech dana będzie następująca funkcja decyzyjna:

$$f(x) = x_1 + x_2x_3 + x_2x_4 + x_3x_4x_5 \quad (3)$$

tabela prawdy dla tej funkcji ma postać:

x_1	x_2	x_3	x_4	x_5	$f(x)$
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	0
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	0
0	0	1	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	1
0	1	1	0	0	1
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	1
1	0	0	0	1	1
1	0	0	1	0	1
1	0	0	1	1	1
1	0	1	0	0	1
1	0	1	0	1	1
1	0	1	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

Wartości odległości Hamminga między ciągiem przyporządkowanym konkretnej zmiennej x_i a ciągiem reprezentującym $f(x)$, są następujące: $H(x_1) = 7, H(x_2) = 11, H(x_3) = 13, H(x_4) = 13, H(x_5) = 15$.

Spostrzeżenie 2: Ze względu na różne wartości odległości Hamminga między ciągami reprezentującymi zmienne x_i a ciągiem $f(x)$ można przypuszczać, że wartości przyjmowane przez zmienne x_i w różnym stopniu przekładają się na wartość badanej funkcji decyzyjnej $f(x)$.

Definicja 3: Istotnością, bądź priorytetem, zmiennej x_i nazywamy zdolność rzutowania wartości przyjmowanych przez tą zmienną na wartość funkcji decyzyjnej $f(x)$.

Definicja 4: Klasą $c(x_i)$ zmiennej x_i nazywamy niepusty zbiór zmiennych funkcji $f(x)$, zawierający zmienną x_i i ewentualnie inne zmienne o takiej samej istotności (priorytecie) jak x_i .

Jak wynika z definicji 4 n -elementowy zbiór $Z_x = \{x_1, x_2, \dots, x_n\}$ zmiennych analizowanej funkcji decyzyjnej można podzielić na podzbiory C_j , za pomocą możliwej do

zdefiniowania relacji równoważności, zawierające zmienne należące do tej samej klasy $c(x_i)$. Tenże podział P można przedstawić następująco:

$$P = \{C_1, C_2, \dots, C_m\}, \text{ gdzie } m < n$$

$$C_j \subset Z_x, \text{ przy } j = \overline{1, m} \quad (4)$$

$$\exists_{\substack{k, r \in \{1, m\} \\ k \neq r}} C_k \cap C_r = \emptyset \text{ oraz } \bigcup_j C_j = Z_x$$

Zmienne znajdujące się w pewnej klasie $c(x_i)$ mają taką samą istotność. W związku z tym niech klasa również posiada swoją istotność – dziedziczoną po zmiennych należących do niej. Istotność klasy $c(x_i)$, a także priorytet zmiennych do niej należących, wyrażona jest wskaźnikiem $w(c(x_i))$ wyznaczanym przy pomocy procedury 1.

Procedura 1: Wykorzystując spostrzeżenie 1. można podać następujący algorytm wyznaczania klas $c(x_i)$ zmiennych funkcji decyzyjnej $f(x)$:

- 1) Na podstawie tabeli prawdy dla każdej zmiennej x_i obliczyć wartość $H(x_i)$.
- 2) Pogrupować zmienne o takiej samej wartości $H(x_i)$ – zmienne te utworzą klasy.
- 3) Posortować klasy (czyli zbiory C_j zawierające zmienne) rosnąco według wartości $H(x_i)$.
- 4) Posortowanym klasom przypisywać wartość wskaźnika $w(c(x_i))$ rozpoczynając od wartości 1 poprzez kolejne liczby całkowite dodatnie.

Przykład 2: Wyznamy klasy zmiennych używając algorytmu opisanego w procedurze 1, dla zadania rozpoczętego w przykładzie 1.

Ad. 1) Wartości $H(x_i)$ zostały już obliczone: $H(x_1) = 7, H(x_2) = 11, H(x_3) = 13, H(x_4) = 13, H(x_5) = 15$.

Ad. 2) Utworzenie zbiorów klas: $C_1 = \{x_1\}, C_2 = \{x_2\}, C_3 = \{x_3, x_4\}, C_4 = \{x_5\}$.

Ad. 3) Posortowanie zbiorów klas (w tym zadaniu są już posortowane):

1. $C_1 = \{x_1\}$ dla $H(x_1) = 7$
2. $C_2 = \{x_2\}$ dla $H(x_2) = 11$
3. $C_3 = \{x_3, x_4\}$ dla $H(x_3) = H(x_4) = 13$
4. $C_4 = \{x_5\}$ dla $H(x_5) = 15$.

Ad. 4) Przypisanie klasom wartości wskaźnika $w(c(x_i))$:

- $w(c(x_1)) = 1$ dla zbioru C_1 ,
- $w(c(x_2)) = 2$ dla zbioru C_2 ,
- $w(c(x_3, x_4)) = 3$ dla zbioru C_3 ,
- $w(c(x_5)) = 4$ dla zbioru C_4 .

W ten sposób zmienne funkcji $f(x)$ zostały przydzielone do czterech klas, gdzie najwyższy priorytet (najbardziej intensywny wpływ na

podjęmowaną decyzję) mają zmienne z klasy pierwszej (przy $w(c(x_1)) = 1$), a najmniejszy z klasy czwartej (przy $w(c(x_5)) = 4$).

3. Metoda określania przybliżonych klas istotności zmiennych

Metoda wyznaczania klas zmiennych za pomocą odległości Hamminga jest metodą skuteczną, choć złożoną obliczeniowo: trzeba wygenerować tablicę prawdy, która posiada 2^n wierszy (gdzie n to liczba zmiennych decyzyjnych) oraz $(n + 1)$ kolumn, a następnie obliczyć wartość miary Hamminga dla n par ciągów o długości 2^n . W związku z tym metoda ta ma złożoność rzędu wykładniczego.

Można zaproponować inną metodę klasyfikacji zmiennych – mniej dokładną, lecz dużo szybszą (obliczeniowo zazwyczaj mniej złożoną, ponieważ nie ma w niej konieczności generowania ani przeglądania tabeli prawdy, a jedynie analizuje się zapis funkcji decyzyjnej $f(x)$ w postaci minimalnej formuły sumacyjnej).

Badając funkcję logiczną $f(x)$ można zauważyć pewne cechy jej zmiennych. W przypadku zmiennych x_i – funkcji decyzyjnej $f(x)$ zgodnej z założeniem 1 – można wyróżnić ich dwie własności: stopień zależności oraz powtarzalność zmiennej.

Definicja 5: Stopień zależności zmiennej to wartość będąca najmniejszą liczbą innych zmiennych występujących w jednym iloczynie wraz z badaną zmienną, w zapisie funkcji $f(x)$.

Definicja 6: Powtarzalność zmiennej to liczba iloczynów, w których w funkcji $f(x)$ występuje badana zmienna.

Ze względu na różne stopnie zależności oraz różną powtarzalność zmiennych tej samej funkcji można przypuszczać, że wartości przyjmowane przez zmienne x_i w różnym stopniu rzutują na wartość badanej funkcji decyzyjnej $f(x)$.

Definicja 7: Przybliżoną klasą $pc(x_i)$ zmiennej x_i nazywamy niepusty zbiór zmiennych funkcji $f(x)$, zawierający zmienną x_i i ewentualnie inne zmienne, które są równoważne z x_i ze względu na taką samą powtarzalność oraz równy stopień zależności.

Spostrzeżenie 3: Jeżeli dana jest funkcja decyzyjna $f(x)$, zgodna z założeniem 1, to można dla zmiennych tej funkcji wyznaczyć przybliżone klasy $pc(x_i)$ grupując zmienne względem stopnia ich zależności, a następnie, wewnątrz każdej takiej grupy, wprowadzając co najmniej jedną klasę $pc(x_i)$ kierując się kryterium powtarzalności zmiennych.

Ważne jest żeby zachować kolejność opisaną w spostrzeżeniu 3: najpierw zmienne są grupowane według stopnia zależności, a dopiero potem według kryterium ich powtarzalności. Jest to istotne ponieważ kryterium zależności jest silniejsze od powtarzalności (uzasadnienie 1).

Uzasadnienie 1: W zapisie sumacyjnym rozważanej funkcji decyzyjnej $f(x)$ zmienne x_i oraz wartość funkcji przyjmują jedynie wartości $\{0,1\}$. Silniejszy wpływ stopnia zależności zmiennej na wartość funkcji można pokazać na przykładzie: niech dana będzie logiczna funkcja decyzyjna $f(x)$, gdzie $x = (x_1, x_2, \dots, x_n)$, $x_i \in \{0,1\}$ i $n \rightarrow \infty$, o postaci:

$$f(x) = x_1 + x_2x_3 + x_2x_4 + x_2x_5 + \dots + x_2x_n \quad (5)$$

gdzie zmienna x_1 występuje tylko jeden raz, ale jest niezależna (jej stopień zależności wynosi zero); zmienna x_2 występuje $(n - 2)$ razy przy $n \rightarrow \infty$, więc liczba jej powtórzeń również dąży do nieskończoności, ale zawsze jest zależna od innej zmiennej x_j (gdzie $j = 3, 4, \dots, n$). Zmienna x_1 ma silniejszy wpływ na wartość funkcji $f(x)$ niż zmienna x_2 ponieważ przyjęcie przez zmienną x_1 wartości „1” gwarantuje, że $f(x) = 1$ – zupełnie niezależnie od wartości zmiennych x_2, x_3, \dots, x_n (w przypadku gdy $x_1 = 0$ wartość $f(x)$ będzie również zależna od pozostałych zmiennych). Natomiast przyjęcie wartości „0” lub „1” przez zmienną x_2 nie jest w stanie zagwarantować (niezależnie od wartości pozostałych zmiennych), że $f(x)$ będzie miała taką samą wartość jak x_2 .

Istotność przybliżonej klasy $pc(x_i)$, a także priorytet zmiennych do niej należących, wyrażona jest przybliżonym wskaźnikiem $pw(pc(x_i))$ wyznaczanym przy pomocy procedury 2.

Procedura 2: Wykorzystując spostrzeżenie 3. można podać następujący algorytm wyznaczania przybliżonych klas zmiennych $pc(x_i)$ funkcji decyzyjnej $f(x)$:

- 1) Zmienne x_i należy pogrupować względem stopnia ich zależności – grupa zmiennych najbardziej niezależnych otrzymuje numer 1, a pozostałe grupy (w kolejności zawierania zmiennych o coraz to wyższym stopniu zależności) są ponumerowane za pomocą kolejnych liczb całkowitych dodatnich (rozpoczynając od liczby 2).
- 2) W każdej grupie, zgodnie z kolejnością numeracji grup z poprzedniego punktu, wyznaczone są przybliżone klasy pc istotności zmiennych x_i : jeżeli grupa zawiera tylko jedną zmienną, to zmienna ta

tworzy osobną przybliżoną klasę; jeżeli grupa zawiera więcej niż jedną zmienną, to przybliżone klasy są wyodrębniane na podstawie kryterium powtarzalności zmiennej. W obrębie jednej grupy zmienne są dzielone na przybliżone klasy według takiej samej powtarzalności – kolejność powstawania przybliżonych klas na tym poziomie jest taka, że najpierw wyodrębniane są klasy składające się ze zmiennych o większej powtarzalności. Kolejno powstającym klasom przybliżonym przypisywana jest wartość wskaźnika $pw(pc(x_i))$ rozpoczynając od wartości 1 poprzez kolejne liczby całkowite dodatnie.

Przykład 3: Oszacujmy przybliżone klasy zmiennych dla funkcji (3) analizowanej w poprzednich przykładach.

Ad. 1) Grupy zależności zmiennych badanej funkcji przedstawiają się następująco:

Numer grupy	Stopień zależności zmiennej	Zmienne wchodzące w skład grupy
1	0	x_1
2	1	x_2, x_3, x_4
3	2	x_5

W pierwszej i trzeciej grupie znajduje się tylko po jednej zmiennej, więc utworzą one osobne przybliżone klasy zmiennych. W drugiej grupie znajdują się trzy zmienne, więc należy sprawdzić jaka jest ich powtarzalność.

Ad. 2) Określenie powtarzalności zmiennych wewnątrz grup zależności:

Wskaźnik pw istotności przybliżonej klasy	Powtarzalność zmiennej *	Zmienne wchodzące w skład przybliżonej klasy pc
1	-	x_1
2	2	x_2, x_3, x_4
3	-	x_5

* Jeżeli zmienna występowała sama w grupie zależności, to liczba jej powtórzeń w sumie iloczynów funkcji $f(x)$ nie jest istotna w wyznaczaniu klas istotności zmiennych.

Porównując otrzymane w powyższym przykładzie wyniki z wynikami z przykładu 2, gdzie klasy były wyznaczane za pomocą odległości Hamminga, widać, że przybliżona klasa o priorytecie „2” (tam gdzie $w_p(pc(x_2, x_3, x_4)) = 2$) powinna być podzielona na dwie osobne klasy $c(x_i)$. Niemniej jednak prawdą jest, że zmienne z tej przybliżonej klasy mają mniejszy wpływ na wartość funkcji decyzyjnej

niż zmienne z przybliżonej klasy o priorytecie „1”, ale jednocześnie mocniej rzutują na wartość funkcji decyzyjnej niż zmienne z przybliżonej klasy o priorytecie równym „3”.

Istnieją także funkcje decyzyjne $f(x)$, zgodne z założeniem 1, takie, że wyznaczanie priorytetów zmiennych za pomocą procedury 1 i procedury 2 daje identyczne wyniki, co pokazuje przykład 4.

Przykład 4: Niech dana będzie funkcja decyzyjna czterech zmiennych:

$$f(x) = x_1 x_4 + x_2 + x_3 \quad (6)$$

Proc. 1:

Ad. 1) Wartości $H(x_i)$ są następujące: $H(x_1) = 7$, $H(x_2) = 5$, $H(x_3) = 5$, $H(x_4) = 7$.

Ad. 2) Utworzenie zbiorów klas: $C_1 = \{x_1, x_4\}$, $C_2 = \{x_2, x_3\}$.

Ad. 3) Sortowanie zbiorów klas:

- $C_2 = \{x_2, x_3\}$ dla $H(x_2) = H(x_3) = 5$
- $C_1 = \{x_1, x_4\}$ dla $H(x_1) = H(x_4) = 7$.

Ad. 4) Przypisanie klasom wartości wskaźnika $w(c(x_i))$:

$$w(c(x_2, x_3)) = 1 \text{ dla zbioru } C_2,$$

$$w(c(x_1, x_4)) = 2 \text{ dla zbioru } C_1.$$

Proc. 2:

Ad. 1) Grupy zależności zmiennych badanej funkcji to:

Numer grupy	Stopień zależności zmiennej	Zmienne wchodzące w skład grupy
1	0	x_2, x_3
2	1	x_1, x_4

Ad. 2) Określenie powtarzalności zmiennych wewnątrz grup zależności:

Wskaźnik pw istotności przybliżonej klasy	Powtarzalność zmiennej	Zmienne wchodzące w skład przybliżonej klasy pc
1	1	x_2, x_3
2	1	x_1, x_4

Przybliżone klasy pokrywają się w tym przykładzie z grupami zależności, ponieważ każda zmienna w zapisie funkcji (6) występuje taką samą liczbę razy.

Jak obrazuje przykład 4., nie zawsze użycie metody mniej złożonej obliczeniowo daje mniej precyzyjny wynik obliczeń.

4. Podsumowanie

W artykule zostały przedstawione dwie metody określania priorytetów zmiennych pewnych funkcji logicznych. W ramach podsumowania spróbujemy odpowiedzieć na pytanie: dlaczego metoda opisana procedurą 1. jest bardziej dokładna niż metoda opisana procedurą 2.?

W procedurze 1. wyznaczanie priorytetów zmiennych opiera się na obliczaniu odległości Hamminga między ciągami reprezentującymi wartości przyjmowane przez każdą zmienną badanej funkcji a ciągiem wartości tejże funkcji. Zatem w każdym pojedynczym kroku badania odmienności konkretnego ciągu x_i i ciągu $f(x)$, bieżąca wartość zmiennej decyzyjnej konfrontowana jest z jedną wartością funkcji, na którą to wartość mają wpływ również wartości pozostałych zmiennych badanej funkcji. Oznacza to, że wyznaczanie wartości $H(x_i)$ dla konkretnej zmiennej x_i nie odbywa się w izolacji od wpływu innych zmiennych. Przeciwna sytuacja ma miejsce w przypadku metody opisanej procedurą 2. – tam określamy priorytet każdej zmiennej na podstawie liczby iloczynów, w których w zapisie funkcji się ona pojawia, oraz najmniejszej liczby innych zmiennych występujących wraz z nią w jednym iloczynie. Analizujemy więc tylko licznosc innych zmiennych, będących w relacji z badaną zmienną, pomijając informacje dotyczące ich cech własnych (takich jak stopień zależności czy powtarzalność), czyli wpływu tychże zmiennych na wartość funkcji decyzyjnej.

5. Bibliografia

- [1] A.V. Aho, J.D. Ullman, *Wykłady z informatyki z przykładami w języku C*, Helion, Gliwice, 2003.
- [2] G. Birkhoff, S. Mac Lane, *Przegląd algebry współczesnej*, PWN, Warszawa, 1960.
- [3] M. Chudy, *Elementy teoretycznych podstaw informatyki*, Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2006.
- [4] T. Łuba, B. Zbierchowski, *Układy logiczne*, Wydawnictwo WIT, Warszawa, 2002.
- [5] A. Mostowski, M. Stark, *Elementy algebry wyższej*, PWN, Warszawa, 1965.
- [6] R. Neapolitan, K. Naimipour, *Podstawy algorytmów z przykładami w C++*, Helion, Gliwice, 2004.
- [7] Praca zbiorowa, *Encyklopedia Universalis. Świat nauki współczesnej. Tom 1*, PWN, Warszawa, 1996.

