Recenzje:
Prof. zw. dr hab. inż. Jacek Migdałek
Dr hab. Paweł Lula

Projekt okładki:
Joanna Sroka

Adiustacja:
Magdalena Polek

ISBN 978-83-7571-028-1

Na zlecenie:
Krakowskiej Szkoły Wyższej im. Andrzeja Frycza Modrzewskiego
www.ksw.edu.pl

Wydawca:
Krakowskie Towarzystwo Edukacyjne sp. z o.o. – Oficyna Wydawnicza AFM,
Kraków 2008

Sprzedaż prowadzi:
Księgarnia Krakowskiego Towarzystwa Edukacyjnego sp. z o.o.
Kampus Krakowskiej Szkoły Wyższej im. Andrzeja Frycza Modrzewskiego
ul. Gustawa Herlinga-Grudzińskiego 1
30-705 Kraków
tel./faks: (012) 252 45 93
e-mail: ksiegarnia@kte.pl

Skład:
Oleg Aleksejczuk

Druk i oprawa:
Eikon

Spis treści

Wstęp

Oddajemy do rąk Czytelników drugi zbiór artykułów z serii „Informatyka". Kontynuuje on prezentację zainteresowań badawczych pracowników Studium Informatyki KSW i Katedry Informatyki Stosowanej Wydziału Ekonomii i Zarządzania KSW.

W pierwszym zeszycie „Informatyki" (pod red. W.Z. Chmielowskiego i M. Pękali, Kraków 2006) dominowała tematyka związana ze sztuczną inteligencją; ten zawiera zagadnienia bardziej różnorodne.

Pierwsze cztery prace związane są z obliczeniami z zakresu przepływów. Rozpoczynamy dyskusją metod obliczeniowych, by przejść do omówienia modeli symulacji sterowania przepływem fali powodziowej oraz prognozowania zużycia wody w gospodarce przy wykorzystaniu logiki rozmytej.

Następne dwie prace poruszają temat sztucznej inteligencji, a dokładniej – baz wiedzy. Użycie w tytułach terminu filozoficznego „ontologia" („ontologiczny") wskazuje na globalne podejście do przedstawianej wiedzy. Prace prezentują szczególne zastosowanie tego ujęcia w odniesieniu do wyrobów metalowych.

Zagadnienie systemu ekspertowego omawia kolejny artykuł. Praca ta wiąże się tematycznie z jednym z artykułów z poprzedniego zeszytu.

Kolejne dwie prace dotyczą zagadnień teoretycznych. Pierwsza opisuje obliczeniowe aspekty „dobrych" generatorów liczb losowych, podstawowego elementu metod Monte Carlo, stosowanych w wielu dyscyplinach. Druga zajmuje się zagadnieniem podobieństwa, czyli zdefiniowaniem w sposób formalny tego, co wydaje się oczywiste nam, a dla maszyny takie nie jest.

Zeszyt kończą prace poruszające temat baz danych. Jedna poświęcona została zarządzaniu danymi w relacyjnych bazach danych, a druga – przechowywaniu elektronicznych wersji publikacji, które stanowią znaczną część zasobów bibliotecznych, archiwalnych i muzealnych.

Redaktorzy mają nadzieję, że drugi zeszyt pozwoli poznać zainteresowanym Czytelnikom zagadnienia związane z zastosowaniem współczesnej informatyki do rozwiązywania problemów w różnorodnych dziedzinach działalności człowieka.

Wojciech Z. Chmielowski, Maciej Pękala

Joanna Płażek

Paradygmaty programowania równoległego w symulacji przepływów adaptacyjną metodą elementów skończonych

1. Wstęp

Obliczenia dużej skali, do których niewątpliwie można zaliczyć symulację przepływów, należą do obszernego działu twórczych zastosowań środków i narzędzi informatyki do rozwiązywania konkretnych, bardzo złożonych problemów obliczeniowych. Coraz częściej konieczne staje się wykorzystanie środowisk rozproszonych, takich jak klastry stacji roboczych, a także komputerów wieloprocesorowych lub metakomputerów sieciowych. Niestety, rozwój sprzętu wyprzedza znacznie rozwój metod efektywnego wykorzystania optymalizowanych algorytmów oraz środków programowych. Informatycy stają obecnie często przed problemem zaawansowanych zastosowań współczesnych architektur komputerowych. Zastosowania te wymagają zarówno bardzo głębokiej wiedzy na temat sprzętu, oprogramowania, jak i dobrej orientacji w problematyce, której dotyczą. Potwierdza to opinię o konieczności prowadzenia badań interdyscyplinarnych.

Efektywne wykorzystanie możliwości nowoczesnych (równoległych) systemów komputerowych wymaga podejmowania kilku pozwalających na możliwie wierne odwzorowanie problemu obliczeniowego na architekturę komputera działań, takich jak:

- określenie w problemie obliczeniowym elementów słabo związanych z innymi oraz występującej lokalności danych dla potencjalnych obliczeń równoległych,
- dobór architektury komputerowej do problemu,
- wybór modelu programowania (równoległego),
- wybór algorytmu obliczeniowego oraz jego efektywna implementacja programowa,
- dobór narzędzi programowych.

Powstają więc pytania o to, jaką przyjąć metodę obliczeniową, jak zaprojektować algorytm, jakie struktury danych wybrać, jakie środki programowe wykorzystać, jaki jest zakres stosowalności istniejących narzędzi informatyki oraz – bardziej ogólnie – czy możliwe jest efektywne rozwiązanie postawionego problemu przy dostępnych środkach informatycznych.

2. Symulacja przepływów adaptacyjną metodą elementów skończonych

Metoda elementów skończonych (MES) jest znana i używana od wielu lat. Stosuje się ją intensywnie w różnych dziedzinach nauk podstawowych i technicznych, między innymi w obliczeniach inżynierskich o bardzo dużej skali złożoności. Wymagania aplikacji MES wobec środków informatyki są bardzo wysokie – obliczenia są czasochłonne, z dużym zapotrzebowaniem na pamięć operacyjną.

Symulacja przepływów adaptacyjną metodą elementów skończonych jest ciągle jedną z intensywnie rozwijanych gałęzi metod obliczeniowych mechaniki, o dużym znaczeniu aplikacyjnym. Numeryczna interpretacja tego zagadnienia zastępuje lub uzupełnia kosztowne pomiary przeprowadzane w tunelach aerodynamicznych, znajduje zastosowanie w lotnictwie i w wielu innych działach techniki i technologii.

Ze względu na niestabilność i nieliniowość zjawisk przepływowych ich badanie jest problemem bardzo złożonym, w którym można wyróżnić kilka wątków. Pierwszy to zaawansowane badania teoretyczne dotyczące modeli matematycznych procesów oraz metod dyskretyzacji czasowej i przestrzennej. Drugi kierunek działalności jest związany z algorytmami obliczeniowymi (zwłaszcza iteracyjnymi), zakresem ich stosowalności, jednoznaczności, zgodności, dokładności, zbieżności i złożoności. Kolejny wątek to komputerowa implementacja algorytmów z uwzględnieniem cech charakterystycznych architektury komputerowej oraz środowisk programowania. Istotnymi zagadnieniami są także: przygotowanie danych wejściowych, czyli siatki obliczeniowej, jej dekompozycja (najczęściej domenowa) dla potrzeb obliczeń równoległych oraz wizualizacja i interpretacja wyników. Ze względu na dużą złożoność problemu często dokonuje się podziału zagadnienia obliczeniowego na kilka dużych elementów, np. oddzielnie opracowuje się i bada generatory siatek pokrywające obszar obliczeniowy lub testuje metody rozwiązania wielkich układów równań z macierzą rzadką. Nie istnieje jedna uniwersalna metoda rozwiązania układu równań liniowych – spośród istniejącego zbioru należy wybrać tę odpowiadającą cechom danego układu równań, wynikającym z charakterystyki symulowanego procesu oraz architektury komputera, dla którego jest ona przeznaczona.

Równoległa realizacja obliczeń nie tylko daje możliwość ich przyspieszenia, ale także, zwłaszcza w przypadku klastrów komputerów, pozwala korzystać z większych zasobów pamięci, co w praktyce oznacza możliwość rozwiązywania problemów o znacznie większych rozmiarach, niż byłoby to możliwe na maszynach z pamięcią wspólną.

Znacznych nakładów obliczeniowych wymaga nie tylko sama symulacja, ale także wizualizacja wielkich plików danych. Równoległość obliczeń wykorzystywana jest do wizualizacji wyników, do ich animacji, jak również do powiązania procesu wizualizacji ze sterowaniem przebiegiem symulacji.

Mimo rozwoju wielu efektywnych algorytmów numerycznej mechaniki płynów grupa zagadnień, które można rozwiązać na dostępnym obecnie sprzęcie komputerowym, jest oczywiście ograniczona. Otrzymanie wiarygodnych wyników symulacji przepływów wokół rzeczywistych obiektów wymaga ogromnej liczby zmiennych modelu dyskretnego, a co za tym idzie – ogromnych nakładów pamięci i mocy obliczeniowej.

3. Implementacja równoległa algorytmu symulacji przepływu

Do zrównoleglenia obliczeń w programie do symulacji przepływów adaptacyjną metodą elementów skończonych można wykorzystać zarówno dwa podstawowe modele programowania równoległego, czyli model z wymianą komunikatów i model z równoległością na poziomie danych, jak i model heterogeniczny oparty na obu wymienionych wyżej modelach. Każdy z tych trzech modeli wymaga innej implementacji i użycia innych środowisk programowania.

3.1. Model z wymianą komunikatów

W modelu z wymianą komunikatów, zwanym również modelem jawnym, odrębne procesy współpracują ze sobą w celu rozwiązania danego problemu. Współpraca ta polega na opartej na przesyłaniu komunikatów wymianie danych między procesami. Model ten wykorzystuje najpopularniejszy schemat programowania rozproszonego *master-slave* (*farmer-worker*). Zostaje w nim wyróżniony jeden program zarządzający (*master*), uruchamiający wszystkie pozostałe procesy (*slaves*) i koordynujący ich pracę, a także zarządzający wejściem/wyjściem programu. Do realizacji tego modelu wykorzystano gotowe narzędzia programowe, takie jak PVM lub MPI. Opracowano jeden program, w którym wykorzystując opcje preprocesora, można dołączyć różne biblioteki i uruchomić program w środowisku albo PVM, albo MPI. Wszystkie utworzone w nich procesy uru-

chamiane są na różnych procesorach komputera wieloprocesorowego lub stacjach roboczych połączonych siecią (stacje mogą być różnych typów i pracować pod różnymi systemami operacyjnymi). PVM (lub MPI) steruje równoległym wykonaniem programu oraz nadzoruje przesyłanie informacji pomiędzy procesami [5]. Każdy z uruchomionych procesów zarządzanych (*slave*) realizuje ten sam algorytm, operując różnymi danymi (model SPMD – Single Program Multiple Data).

Do głównych zadań procesu zarządzającego (*master*) należą:
- geometryczna dekompozycja obszaru obliczeniowego,
- rozesłanie do poszczególnych jednostek CPU danych i parametrów symulacji związanych z określonym podobszarem,
- opracowanie tablic przesłań niezbędnych do wymiany komunikatów między podobszarami,
- zebranie oraz zapamiętanie wyników realizacji jednego kroku czasowego obliczeń.

Procesy zarządzane (*slaves*):
- tworzą własną strukturę danych na podstawie informacji nadesłanych od procesu *master*,
- realizują obliczenia jednego kroku czasowego, w trakcie którego następuje wymiana komunikatów między procesami,
- wyróżniony proces zbiera informacje od pozostałych, dokonuje agregacji wyników cząstkowych i rozsyła je do wszystkich procesów,
- przesyłają wyniki symulacji do procesu *master*.

3.2. Model z równoległością na poziomie danych

Drugi z modeli to model z równoległością na poziomie danych, zwany również modelem niejawnym. Może być stosowany w komputerach posiadających pamięć wspólną. Polega na zleceniu kompilatorowi utworzenia wersji równoległej programu na podstawie wcześniej przeprowadzonej przez niego analizy. Można wyróżnić model wykorzystujący opcje kompilatorów lub środowisko OpenMP [3].

Podstawową techniką wykorzystywaną w środowisku OpenMP do zrównoleglenia kodu programu jest użycie dyrektyw kompilatora. Postacie dyrektyw kompilatora na różnych komputerach mogą nie być identyczne. Wynika to stąd, że firmy produkujące takie komputery, jak SGI, Cray czy SUN opracowały niezależnie swoje własne zbiory dyrektyw. Są one jednak bardzo zbliżone pod względem zapisu i pełnionych funkcji. Środowisko OpenMP pozwala ujednolicić postać dyrektyw, a także udostępnia wiele procedur, które wywoływane są podczas wykonywania programu. Dodatkowo OpenMP dostarcza bogatą bibliotekę zawie-

rającą szereg elementów kontrolnych i synchronizujących, które mogą być umieszczane poza wydzielonymi wcześniej obszarami podlegającymi zrównolegleniu.

Zastosowanie dyrektyw kompilatora pozwala użytkownikowi wskazać w programie sekwencyjnym te części, które mają się wykonywać równolegle, oraz określić sposób ich zrównoleglenia. Wyróżnione części programu rozkładane są na wątki wykonywane na oddzielnych procesorach. Główną zaletą stosowania dyrektyw jest możliwość użycia ich na platformie zarówno jedno-, jak i wieloprocesorowej. Oznacza to, że w przypadku wykonywania programu na jednym procesorze dyrektywy podczas kompilacji są ignorowane.

3.3. Ograniczenia stosowanych modeli

Oba opisane wyżej modele – model z wymianą komunikatów oraz model z równoległością na poziomie danych – mają określone zakresy zastosowań. Pierwszy z nich, model z wymianą komunikatów, wykorzystuje środowiska programowania rozproszonego, takie jak PVM czy MPI. Główną ich zaletą jest to, że można z nich korzystać w przypadku:

- sieci stacji roboczych:
  - homogenicznej (gdy stacje robocze są identyczne pod względem wydajności obliczeniowej),
  - heterogenicznej (gdy występują różnice mocy obliczeniowej pomiędzy poszczególnymi stacjami),
- komputera wieloprocesorowego,
- klastra.

Szeroki zakres stosowania modelu z wymianą komunikatów wynika stąd, że może on współpracować zarówno z pamięcią wspólną jak i z pamięciami rozproszonymi pomiędzy kilkoma maszynami. Oczywiście w przypadku komputera wieloprocesorowego komunikacja między procesami jest znacznie szybsza niż w przypadku sieci stacji roboczych. Również zapewnienie równomiernego obciążenia poszczególnych procesorów jest łatwiejsze do uzyskania dzięki ich identycznym parametrom. Wykonanie obliczeń na heterogenicznej sieci stacji roboczych wymaga dostosowania rozmiaru i wielkości nakładu obliczeniowego procesów do mocy poszczególnych procesorów. Główną zaletą zespołów stacji roboczych jest dostęp do pamięci o dużo większym rozmiarze niż w przypadku maszyn wieloprocesorowych, co daje możliwość rozwiązywania problemów, które ze względu na swój duży rozmiar nie mogą być symulowane na komputerach z pamięcią wspólną, a także umożliwia utworzenie wirtualnego komputera sieciowego praktycznie bez nakładów.

Drugi z modeli, model z równoległością na poziomie danych, jest ściśle związany z nowoczesnymi architekturami i metodami kompilacji, bazuje na systemach

z pamięcią wspólną. Wszystkie wykorzystywane w tym modelu procesory korzystają ze wspólnej pamięci operacyjnej, co zapewnia szybką komunikację między nimi. Niestety zrównoleglenie w programie jedynie wybranych przez nas konstrukcji nie daje tak znacznego przyspieszenia obliczeń jak w przypadku modelu z przesyłaniem komunikatów, w którym dużo większe fragmenty programu wykonywane są równolegle.

Z powodów opisanych powyżej zaistniała konieczność opracowania dla architektur z wyróżnionymi grupami procesorów modelu heterogenicznego, łączącego oba opisane modele, wykorzystującego zarówno pamięć globalną, jak i pamięci lokalne poszczególnych węzłów wieloprocesorowych.

3.4. Model heterogeniczny

Biorąc pod uwagę ograniczenia omawianych poprzednio modeli, zaproponowano stworzenie modelu heterogenicznego obliczeń równoległych. Pozwala on na równoczesne zastosowanie modelu z przesyłaniem komunikatów i modelu z równoległością na poziomie danych. Bazuje na systemie MPI (lub PVM), zapewniającym przesyłanie informacji pomiędzy podobszarami, oraz na mechanizmach zrównoleglających wykonanie pętli, realizowanych przez nowoczesne kompilatory.

W modelu heterogenicznym, zwanym również modelem dwupoziomowego zrównoleglenia obliczeń, podobnie jak w modelu z przesyłaniem komunikatów, proces główny uaktywnia procesy podrzędne [6]. Różnica jednak polega na tym, że w tym przypadku każdy z procesów podrzędnych jest dedykowany na zespół procesorów (węzeł obliczeniowy wieloprocesorowy) połączonych ze sobą szybką warstwą komunikacyjną. Połączenia pomiędzy poszczególnymi procesami (grupami procesorów) są wolniejsze i zapewniają przesyłanie komunikatów pomiędzy podobszarami.

Należy podkreślić, że największy wpływ na czas realizacji programu z zaimplementowanymi modelami z przesyłaniem komunikatów i heterogenicznym ma sposób, w jaki dokonuje się geometrycznej dekompozycji obszaru [1]. Wynika to stąd, że czas obliczeń każdego procesora jest ściśle związany z liczbą węzłów przydzielonej mu podsiatki. Liczba węzłów wewnętrznych podobszaru określa, ile będzie tworzonych łat elementów i rozwiązywanych dla nich układów równań. Ich liczba powinna być uzależniona od mocy procesora wykonującego obliczenia oraz od jego obciążenia i zapewniać jednakowy czas obliczeń dla każdej podsiatki. Natomiast liczba węzłów na granicy podobszarów decyduje o długości komunikatów rozsyłanych pomiędzy podobszarami i powinna przyjmować wartość minimalną. W zaimplementowanej równoległej wersji programu do dekompozycji obszaru wykorzystano algorytm przesuwającego się frontu [4]. Polega on na sukcesywnym dobieraniu do tworzonego podobszaru węzłów spośród tych umieszczonych w tzw. froncie.

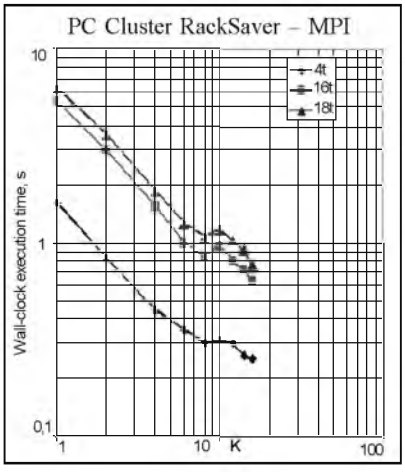
Opisany wyżej model heterogeniczny przeznaczony jest do uruchomienia na nowoczesnych komputerach wieloprocesorowych klasy cc-NUMA, zespołach unixowych stacji roboczych oraz klastrach.

4. Wyniki

Do badań wykorzystano program przeznaczony do symulacji przepływów gazów ściśliwych, nielepkich. Zastosowanym w nim modelem matematycznym są równania Eulera (równania Naviera–Stokesa z pominiętymi wyrażeniami opisującymi efekty lepkie) [2]. Do symulacji wykorzystano siatki wygenerowane wcześniej przez generator siatek trójkątnych, niestrukturalnych, a następnie poddane adaptacji w trakcie realizacji kolejnych kroków czasowych. Do rozwiązywania układu równań zastosowano iteracyjną metodę GMRES ze wstępną poprawą uwarunkowania macierzy.

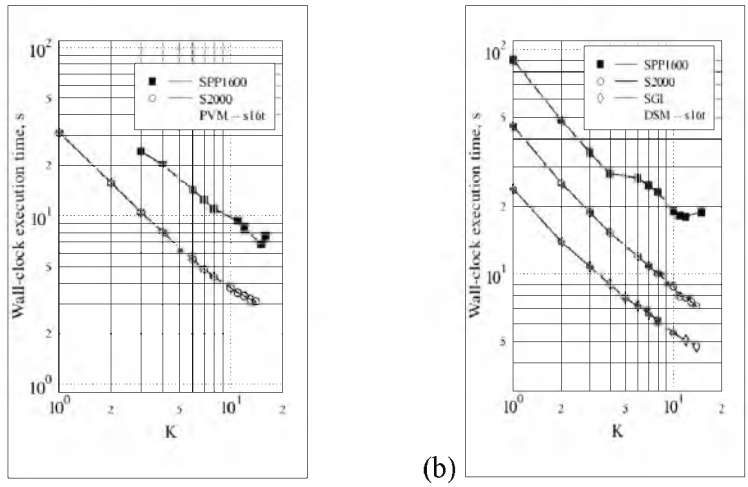
Programy równoległe zostały przetestowane na komputerach SGI Origin2800 i klastrze komputerów RackSaver. Wyniki porównano z uzyskanymi wcześniej wynikami testów przeprowadzonych na komputerach HP Exemplar SPP1600/XA, HP Exemplar S2000 i SGI Origin2000. Przykład opisuje zachowanie fali uderzeniowej o liczbie Macha 10, która napotyka prostopadły do niej klin o kącie rozwarcia 60 stopni.

Rysunek 1 przedstawia czasy symulacji uruchomionych na klastrze RackSaver dla siatek liczących 4474 (4t), 16 858 (16t) oraz 18 241 (18t) węzłów. Testy przeprowadzono dla modelu z wymianą komunikatów w środowisku MPI. Przy początkowym zwiększaniu liczby procesorów widać wyraźne przyspieszenie obliczeń, a następnie opóźnienie związane najprawdopodobniej z architekturą połączeń kolejnych jednostek.



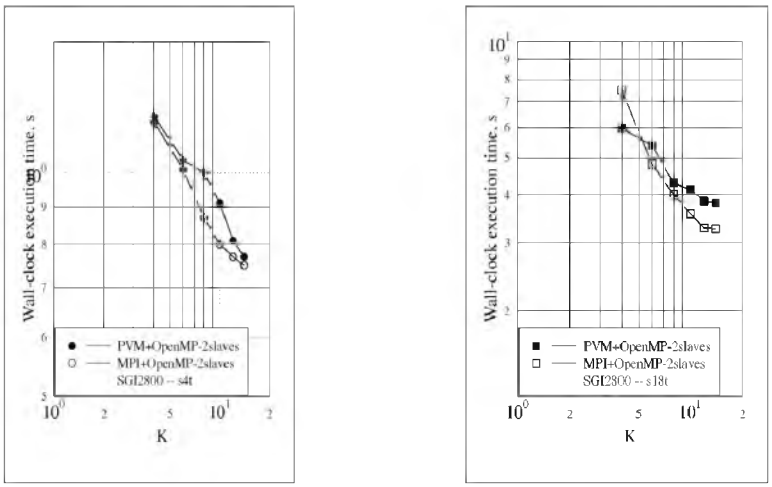
Rysunek 1. Czasy symulacji interakcji fali uderzeniowej z klinem dla siatek 4474 (4t), 16 858 (16t) i 18 241 (18t) węzłów w środowisku MPI

Na rysunku 2 przedstawiono czasy symulacji wykonanych na komputerach HP Exemplar SPP1600/XA, HP Exemplar S2000 i SGI Origin2000. Na rysunku 2a wyniki dotyczą modelu z wymianą komunikatów w środowisku PVM, a na rysunku 2b – modelu z równoległością na poziomie danych wykorzystującego dyrektywy zrównoleglające kompilatora języka C (DSM).



Rysunek 2. Czasy symulacji interakcji fali uderzeniowej z klinem dla siatki 16 858 (16t) węzłów w środowiskach PVM i DSM

Na rysunku 3 przedstawiono czasy symulacji wykonanych na komputerze SGI Origin2800 wykorzystujących model heterogeniczny z zastosowaniem środowisk



Rysunek 3. Czasy symulacji interakcji fali uderzeniowej z klinem dla siatek 4474 (4t) i 18 241 (18t) węzłów w środowiskach PVM + OpenMP i MPI + OpenMP

PVM i OpenMP oraz MPI i OpenMP. W obu przypadkach symulacje przeprowadzono dla dwóch procesów podrzędnych (*slaves*). Jak widać na rysunku, lepsze wyniki uzyskano dla modelu wykorzystującego do przesyłania komunikatów środowisko MPI.

5. Podsumowanie

Przeprowadzone badania potwierdziły celowość zastosowania obliczeń rozproszonych w zadaniach mechaniki płynów. Pokazały, w jaki sposób wybór architektury komputera i modelu programowania równoległego wpływają na efektywność obliczeń. Podsumowując, można stwierdzić, że:

- zaimplementowana metoda elementów skończonych z wykorzystaniem solwerów iteracyjnych nie wymaga tworzenia globalnej macierzy sztywności i dlatego nadaje się szczególnie do obliczeń równoległych wykorzystywanych do rozwiązywania problemów o dużych rozmiarach,
- model z przesyłaniem komunikatów daje największe przyspieszenie obliczeń,
- model z równoległością na poziomie danych jest znacznie łatwiejszy do zaimplementowania niż pozostałe modele,
- model heterogeniczny zaimplementowany na komputerach z hierarchiczną organizacją pamięci pozwala w pełni wykorzystać pamięci z różnych poziomów komputera,
- efektywność obliczeń w środowisku MPI jest większa niż w środowisku PVM.

Bibliografia

[1] Farhat C., Lanteri S., Simon H.D., *TOP/DOMDEC – a Software Tool for Mesh Partitioning and Parallel Processing*, „Journal of Computing Systems in Engineering" 1995, 6, s. 13–26.

[2] Hirsch C., *Numerical Computation of Internal and External Flows*, J. Wiley & Sons, 1998.

[3] *OpenMP – A Parallel Programming Model for Shared Memory Architectures*, Edinburgh 1998.

[4] Płażek J., Banaś K., Kitowski J., *Exploiting Two-Level Parallelism in FEM Applications*, [w:] B. Hertzberge, P. Sloot (eds.), *Proceedings of the International Conference and Exhibition on High-Performance Computing and Networking, 1997 April 28–30, 1997*, „Lecture Notes in Computer Science" 1997, vol. 1225, s. 272–281.

[5] Płażek J., Banaś K., Kitowski J., *Comparison of Message-Passing and Shared Memory Implementations of the GMRES Method on MIMD Computers*, „Scientific Programming" 2001, 9(4), s. 195–209.

[6] Płażek J., Jurczyk T., Kitowski J., *Comparison of Hybrid Programming Models for the GMRES Method*, Timisoara 2003, s. 236–243.

Wojciech Z. Chmielowski

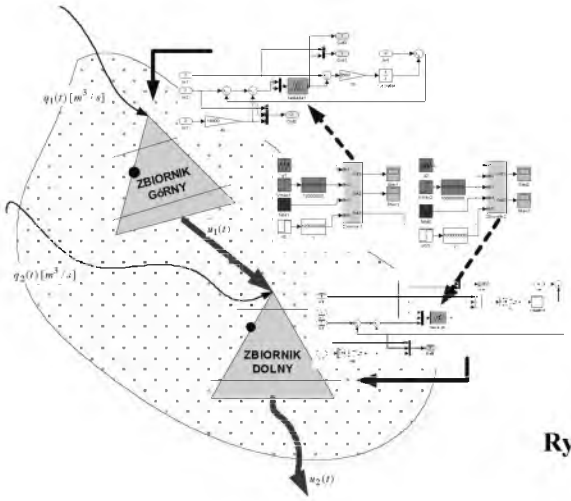
Regulatory rozmyte sterujące przejściem fali powodziowej przez kaskadę zbiorników retencyjnych

1. Wstęp

Regulatory rozmyte (*fuzzy sets*) oraz symulacyjny układ sterujący przejściem fali powodziowej przez kaskadę zbiorników retencyjnych skonstruowano w środowisku Matlab z zastosowaniem modułu (Toolbox) Fuzzy.

Konstrukcję regulatorów rozpoczęto od zdefiniowania, oznaczenia i określenia tzw. FIS (*fuzzy interface system*). Następnie zdefiniowane regulatory rozmyte zastosowano w układzie regulacji. Układ regulacji powstał w module Matlab/Simulink. Symulację sterowania falą powodziową przez hipotetyczną kaskadę zbiorników przeprowadzono, opierając się na danych syntetycznych, dostrojenia parametrów regulatora dokonano z wykorzystaniem AG (algorytmu genetycznego). Otrzymany w ten sposób układ sterowania można testować historycznymi falami powodziowymi zaistniałymi na wybranych zbiornikach, np. w dorzeczu górnej Wisły.

Otrzymane wyniki są bardzo obiecujące, co świadczy o niezwykłej skuteczności działania układów deskryptywnego sterowania rozmytego.



Rysunek 1. Kaskada zbiorników

2. Zbiornik górny – FIS (*fuzzy interface system*)

Wektor zmiennych wejściowych został zdefiniowany następująco:

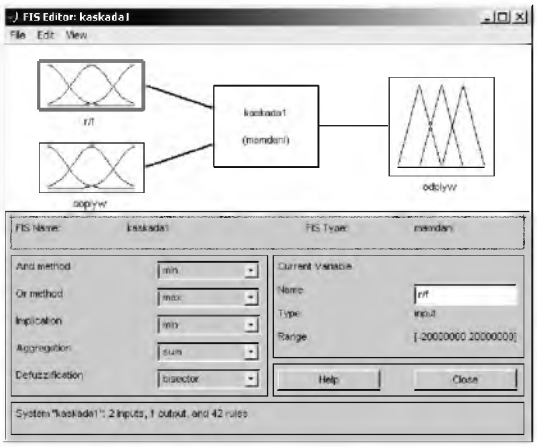
$$X(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^T \tag{1}$$

w którym:

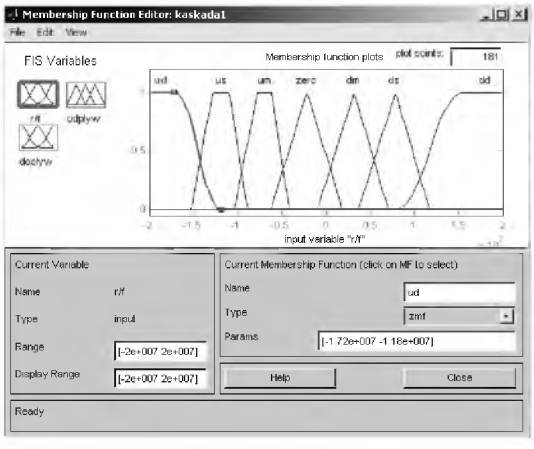
$x_1(t)$ różnica wynikająca z wolnej objętości zbiornika, a prognozowanej objętości fali powodziowej na najbliższe 4 godziny $[\mathrm{m}^3]$; wolna objętość to różnica między pojemnością maksymalną zbiornika a jego chwilowym stanem $\forall t \in [0,T]$;

$x_2(t)$ rzeczywisty dopływ do zbiornika $[\mathrm{m}^3/\mathrm{s}]$, $\forall t \in [0,T]$.

Dla wektora $X(t) = \begin{bmatrix} x_1(t) & x_2(t) \end{bmatrix}^T$ przyjęto uniwersum: $U^T = \begin{bmatrix} u_1 & u_2 \end{bmatrix}$, $u_1 = \begin{bmatrix} -2 \cdot 10^7 & 2 \cdot 10^7 \end{bmatrix}$, $[\mathrm{m}^3]$ oraz $u_2 = \begin{bmatrix} 0 & 5000 \end{bmatrix}$, $[\mathrm{m}^3/\mathrm{s}]$.



Rysunek 2. FIS (*fuzzy interface system*). Edytor zmiennych wejściowych, bazy reguł, zmiennych wyjściowych dla zbiornika górnego



Rysunek 3. Uniwersum i zbiór termów dla zmiennej $x_1(t)$ (wolna objętość zbiornika)

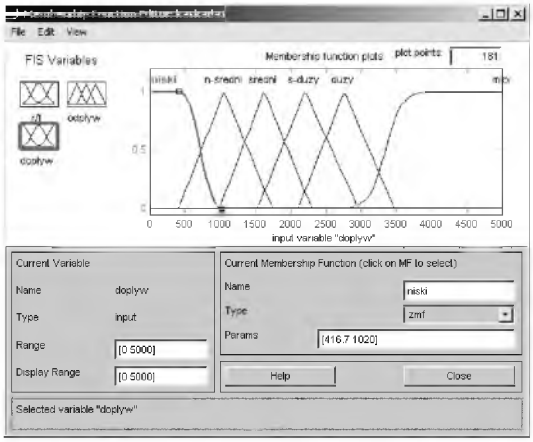
Wektor termów T_{x1} dla zmiennej wejściowej $x_1(t)$ (rysunek 3) zdefiniowano jak niżej:

$$T_{x1} = [t_{x1,1} = ud \quad t_{x1,2} = us \quad t_{x1,3} = um \quad t_{x1,4} = zero \quad t_{x1,5} = dm \quad t_{x1,6} = ds \quad t_{x1,7} = dd] \tag{2}$$

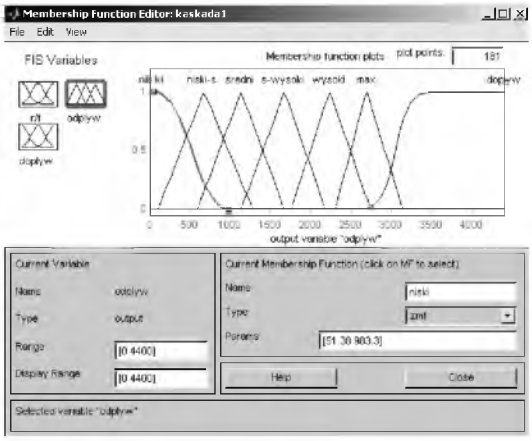
Każdy z termów (zbiorów rozmytych) to liczba rozmyta o podanym kształcie i parametrach. Odpowiedni edytor umożliwia wybranie kształtu, parametrów oraz umiejscowienie termu na uniwersum.

Wektor termów T_{x2} dla zmiennej wejściowej $x_2(t)$ (rysunek 4) zdefiniowano według tej samej zasady:

$$T_{x2} = \left[t_{x2,1} = niski \quad t_{x2,2} = n - średni \quad t_{x2,3} = średni \quad t_{x2,4} = s - duży \quad t_{x2,5} = duży \quad t_{x2,6} = max. \right] \tag{3}$$



Rysunek 4. Uniwersum i zbiór termów dla zmiennej $x_2(t)$ (dopływ do zbiornika)



Rysunek 5. Uniwersum i zbiór termów dla zmiennej wyjściowej $y_1(t)$ (odpływ)

Zmienną wyjściową (rysunek 5) zdefiniowano następująco:
$y(t)$ określony przez regulator rozmyty odpływ ze zbiornika $[m^3/s]$, $\forall t \in [0,T]$.

Dla zmiennej $y(t)$ przyjęto uniwersum $u_1 = [0\ 4400]$, $[m^3/s]$.

Wektor termów T_{y1} dla zmiennej wyjściowej $y_2(t)$ zdefiniowano jako:

$$T_{y1} = \begin{bmatrix} t_{y1,1} = niski & t_{y1,2} = n - średni & t_{y1,3} = średni \\ t_{y1,4} = s - wysoki & t_{y1,5} = wysoki & t_{y1,6} = max. \quad t_{y1,7} = dopływ \end{bmatrix} \quad (4)$$

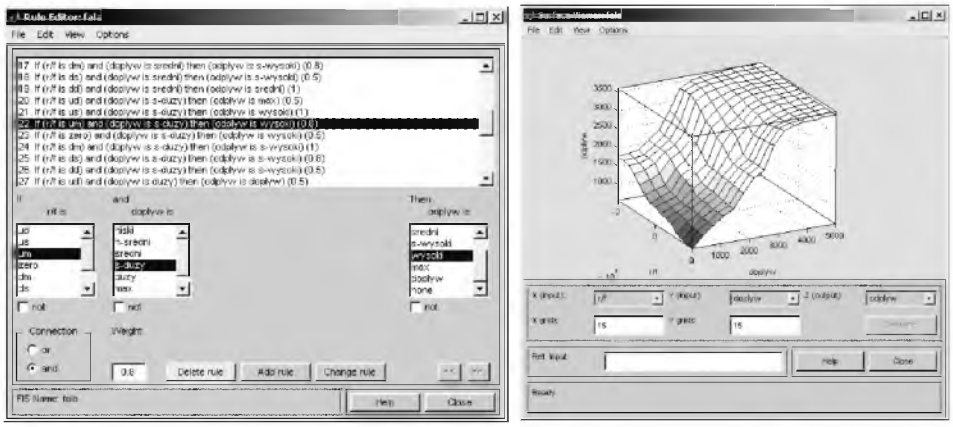
Baza reguł (rysunek 6) zawiera 42 zależności logiczne między zmiennymi wejściowymi $X(t) = [x_1(t) \quad x_2(t)]^T$ a zmienną wyjściową $y(t)$.

Jest to z pewnością najtrudniejsza do zaprojektowania część regulatora rozmytego. Regulator wnioskuje na podstawie informacji zawartych w bazie reguł. Reguły są odzwierciedleniem znajomości sterowanego procesu, wiedzy oraz doświadczenia osoby projektującej bazę reguł. Jak zaznaczono [10], deskryptywne sterowanie rozmyte stanowi pewien system ekspercki, w którym zakłada się, że:

- nie dysponujemy modelem sterowanego procesu lub z uwagi na jego złożoność, skomplikowaną formę, nieliniowość itp. nie chcemy lub nie możemy go zastosować w procesie sterowania;
- dysponujemy instrukcjami dyspozytorskimi, wiedzą, doświadczeniem, intuicją instruktorów, decydentów, dzięki czemu posiadamy informacje o tym, jak sterować procesem bez znajomości jego modelu;
- wyżej wymienioną wiedzę, intuicję i doświadczenia operatorzy systemu (decydenci) mogą przekazać jedynie w postaci naturalnej (werbalnej), a nie w postaci dokładnej, np. numerycznej.

Niewłaściwe zestawienie bazy reguł wynikające z braku dostatecznej wiedzy na temat sterowanego procesu prowadzi w konsekwencji do złego wnioskowania i często absurdalnych wyników.

Należy również podkreślić, że na potrzeby zaprezentowania modelu symulacyjnego przyjęto w sposób wysoce dowolny zakres uniwersum poszczególnych zmiennych wejściowych i wyjściowych. Zastosowanie regulatorów dla konkretnego obiektu sterowania (konkretnych zbiorników) wymaga dostosowania uniwersum poszczególnych zmiennych do fizycznych parametrów obiektów.



Rysunek 6. Baza reguł dla rozpatrywanego zbiornika, powierzchnia decyzyjna wynikająca z bazy reguł

3. Zbiornik dolny – FIS (*fuzzy interface system*)

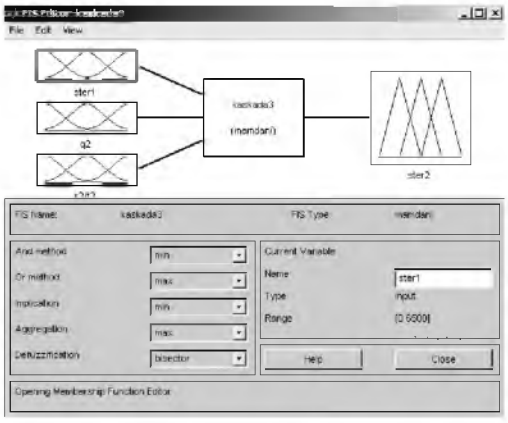
Dla zbiornika dolnego wektor zmiennych wejściowych został zdefiniowany następująco:

$$X(t) = \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) \end{bmatrix}^T \tag{5}$$

w którym:

$x_1(t)$ sterowany odpływ ze zbiornika górnego (kontrolowany dopływ do zbiornika dolnego) $[\mathrm{m}^3/\mathrm{s}]$, $\forall t \in [0,T]$;

$x_2(t)$ dopływ boczny do zbiornika dolnego $[\mathrm{m}^3/\mathrm{s}]$, $\forall t \in [0,T]$;



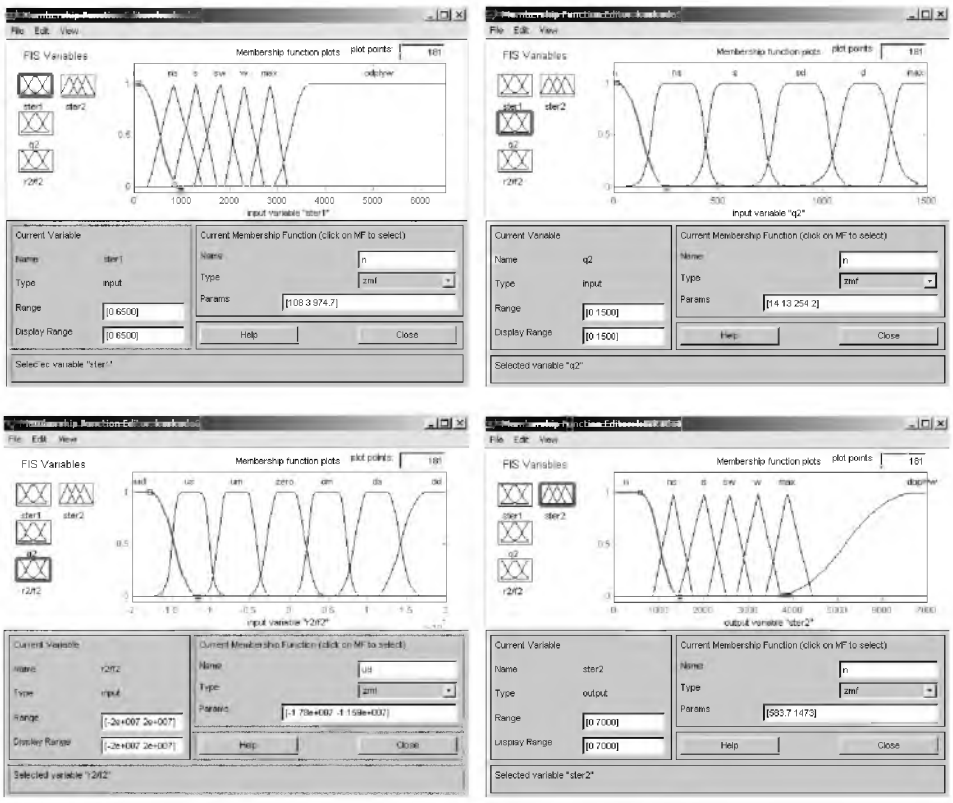
Rysunek 7. FIS (*fuzzy interface system*). Edytor zmiennych wejściowych, bazy reguł, zmiennych wyjściowych dla zbiornika dolnego

$x_3(t)$ różnica wynikająca z wolnej objętości zbiornika dolnego, a prognozowanej objętości fali powodziowej dla tego zbiornika na najbliższe 4 godziny $[\mathrm{m}^3]$; wolna objętość to różnica między pojemnością maksymalną zbiornika a jego chwilowym stanem $\forall t \in [0,T]$.

Dla wektora $X(t) = [x_1(t) \quad x_2(t) \quad x_3(t)]^T$ przyjęto uniwersum:
$U^T = [u_1 \quad u_2 \quad u_3]$, $u_1 = [0 \quad 6500]$, $[m^3/s]$, $u_2 = [0 \quad 1500]$, $[m^3/s]$
oraz $u_3 = [-2 \cdot 10^7 \quad 2 \cdot 10^7]$, $[m^3]$.

Zmienną wyjściową (rysunek 8) zdefiniowano następująco:
$y(t)$ określony przez regulator rozmyty odpływ ze zbiornika dolnego w $[\mathrm{m}^3/\mathrm{s}]$, $\forall t \in [0,T]$.

Dla zmiennej $y(t)$ przyjęto uniwersum: $u_1 = [0 \ 4400]$, $[\mathrm{m}^3/\mathrm{s}]$.



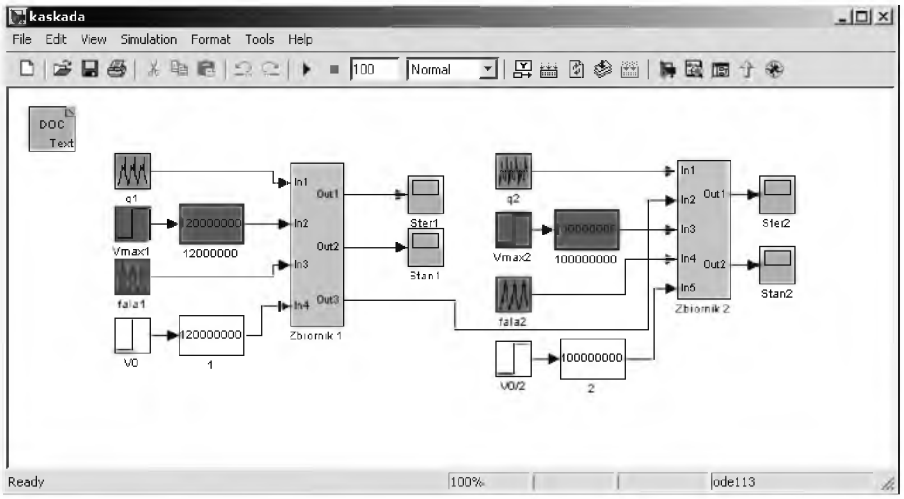
Rysunek 8. Uniwersum i zbiór termów dla zmiennych wejściowych i wyjściowych

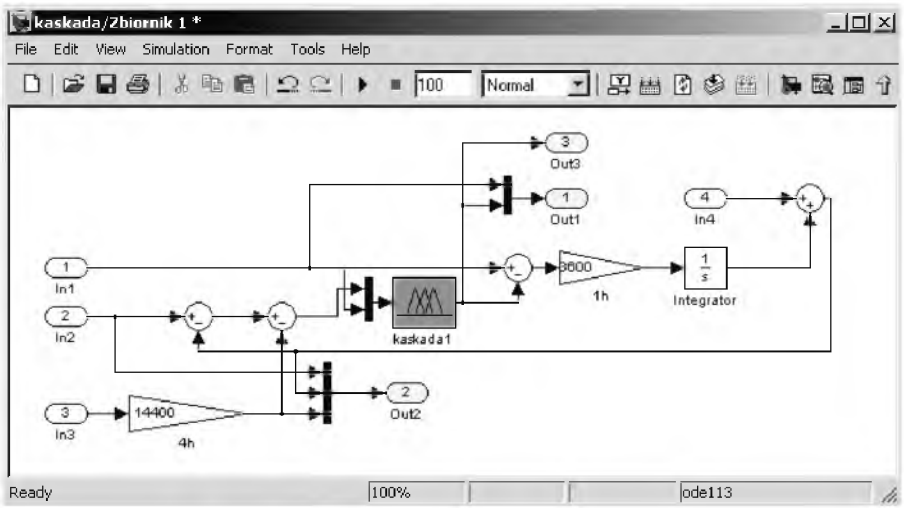
Wektor termów T_{y1} dla zmiennej wyjściowej $y_2(t)$ zdefiniowano jako:

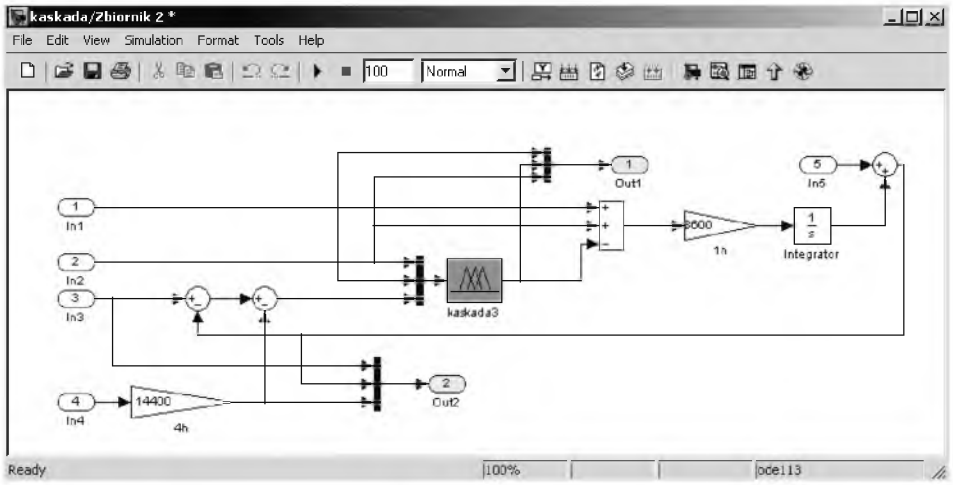
$$T_{y1} = \begin{bmatrix} t_{y1,1} = n & t_{y1,2} = ns & t_{y1,3} = s \\ t_{y1,4} = sw & t_{y1,5} = w & t_{y1,6} = max. \quad t_{y1,7} = dop\l yw \end{bmatrix} \tag{6}$$

Baza reguł zawiera aż **294** zależności logiczne między zmiennymi wejściowymi $X(t) = [x_1(t) \quad x_2(t) \quad x_3(t)]^T$ a zmienną wyjściową $y(t)$.

4. Układ sterowania z zastosowaniem regulatorów rozmytych







Rysunek 9. Układ sterowania z zastosowaniem regulatorów rozmytych

Układ sterowania przejściem fali powodziowej przez hipotetyczną kaskadę zbiorników zrealizowano w module Matlab/Simulink. Danymi wejściowymi do układu są:

- godzinowe dopływy do zbiornika górnego i dolnego $[m^3/s]$,
- prognozowana objętość fali powodziowej na najbliższe 4 godziny dla zbiornika górnego i na dopływie bocznym dla zbiornika dolnego $[m^3]$,
- pojemność początkowego wypełnienia zbiornika górnego i dolnego w chwili rozpoczęcia symulacji (chwili traktowanej jako początek fali powodziowej) $[m^3]$.

W trakcie symulacji (rysunek 10) postawiono bardzo trudne warunki początkowe dla zbiorników:

- zerowa rezerwa powodziowa (zbiorniki całkowicie wypełnione),
- dopływ do zbiornika o bardzo dużych wartościach godzinowych,
- prognozowana objętość każdej fali powodziowej rzędu 50–80% całkowitej objętości zbiorników.

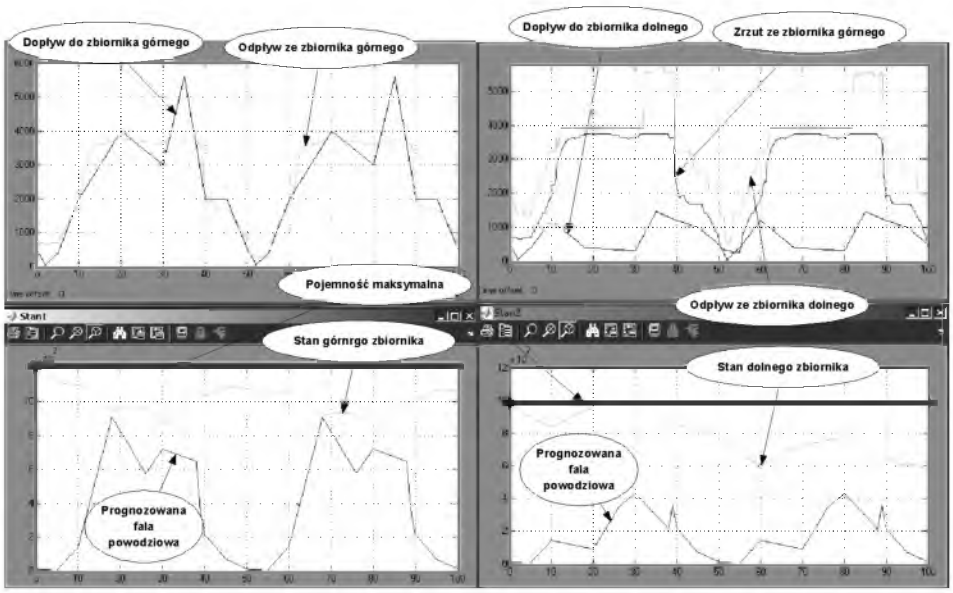
Rysunek 10 (lewa część) przedstawia trajektorie dopływu oraz zadysponowanego przez regulator rozmyty odpływu ze zbiornika górnego, następnie trajektorię uzyskanego w wyniku sterowania stanu zbiornika oraz trajektorie prognozowanych fal powodziowych w okresie obserwacji.

Rysunek 10 (prawa część) przedstawia wyżej wymienione przebiegi w odniesieniu do zbiornika dolnego.

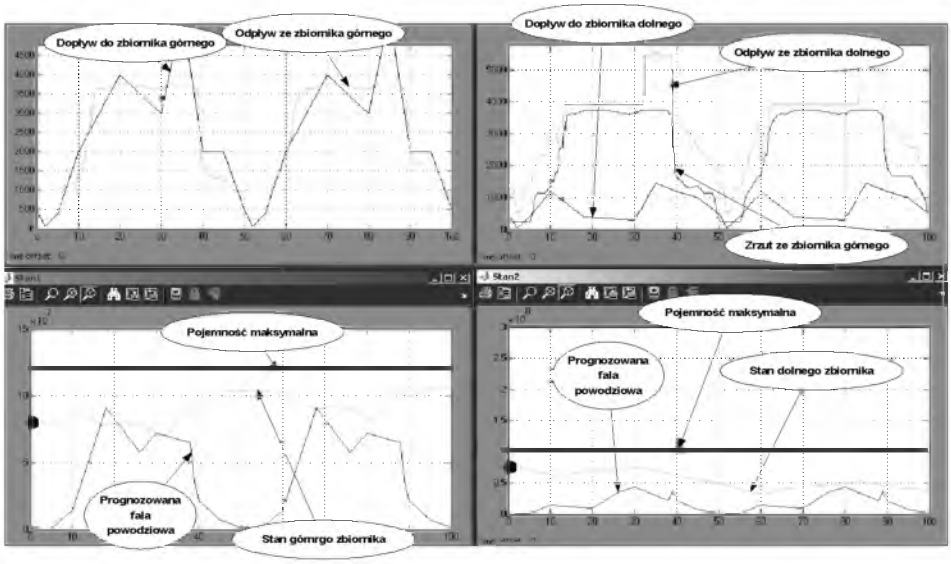
Rezultaty pracy regulatorów są zaskakująco dobre. W wyniku obcinania fal powodziowych trajektorie stanów zbiorników mieszczą się w ograniczeniach i w konsekwencji zastosowanych odpływów następuje stopniowe obniżanie stanów obu zbiorników.

Po przyjęciu mniej drastycznych warunków początkowych (niższe wypełnienia początkowe zbiorników) rezultaty sterowania opartego na regulatorach rozmytych są jeszcze bardziej korzystne.

Dla zadysponowanej fali powodziowej (max. 90 mln m^3, tj. ~80% całkowitej pojemności górnego zbiornika) następuje zupełnie spokojne, gładkie (ze względu na stany zbiorników) przeprowadzenie fali powodziowej przez kaskadę zbiorników (rysunek 11). Dalsze obniżanie stanów początkowych zbiorników poprawia jedynie sytuację bezpiecznego sterowania kaskadą zbiorników w warunkach wezbrania powodziowego.



Rysunek 10. Przejście fali powodziowej przez kaskadę zbiorników



Rysunek 11. Przejście fali powodziowej przez kaskadę zbiorników

5. Podsumowanie

W kolejnych punktach artykułu przedstawiono operacje i przekształcenia, konieczne przy formowaniu i projektowaniu regulatorów rozmytych. W tym konkretnym przypadku były to regulatory sterujące przejściem fali powodziowej przez kaskadę zbiorników retencyjnych.

Ilość zagadnień, problemów, metod i rozwiązań związanych z regulatorami rozmytymi i układami regulacji z ich zastosowaniem jest olbrzymia. Świadczy o tym bardzo obszerna literatura przedmiotu, krajowa i zagraniczna. Doświadczenia i eksperymenty wielu lat (pierwsze prace prof. Lotfiego Zadeha pochodzą z 1967 roku) zaowocowały licznymi rozwiązaniami przemysłowymi na dużą skalę w różnych dziedzinach gospodarki.

Ostatnio w literaturze przedmiotu obserwuje się większe zainteresowanie preskryptywnym sterowaniem rozmytym, w którym zakłada się ścisły algorytm sterowania i nadrzędną funkcję celu, traktowaną jako ocena zastosowanego sterowania. To podejście bliższe jest idei sterowania, które z założenia opiera się na znajomości procesu, celu i wymagań dotyczących sterowania.

Oba kierunki (deskryptywny i preskryptywny) są bardzo interesujące, a w połączeniu z zastosowaniem sieci neuronowych i algorytmów genetycznych stanowią niezwykle silny i nowoczesny aparat w zakresie sterowania procesami technologicznymi i modelowania matematycznego.

Bibliografia

[1] Aoki S., Kwachi S., *Application of Fuzzy Control for Dead-time Processes in a Glass Melting Furnace*, „Fuzzy Sets and Systems" 1990, vol. 38, No. 5, s. 251–256.

[2] Aracil J., Garcia-Cerezo A., Ollero A., *Fuzzy Control of Dynamical Systems. Stability Analysis Based on the Conicity Criterion. Proceedings of the 4th Iternational Fuzzy Systems Association Congress*, Brussels 1991.

[3] Arita S., *Development of an Ultrasonic Cancer Diagnosis System Using Fuzzy Theory*, „Japanese Journal of Fuzzy Theory and System" 1991, vol. 3, No. 3, s. 215–230.

[4] Arita S., Tsutsui T., *Fuzzy Logic Control of Blood Pressure Through Inhalational Anesthesia. Proceedings of the First International Conference on Fuzzy Logic and Neural Networks*, Iizuka 1990.

[5] Babuska R., *Fuzzy Modeling – a Control Engineering Perspective. Proceedings of the International Conference FUZZ-IEEE/IFES'95*, Yokohama 1995.

[6] Cao S.G., Rees N.W., Feng G., *Analysis and Design for a Class of Complex Control System*, part I: *Fuzzy Modeling and Design*, „Automatica" 1997, vol. 33, No. 6, s. 1017–1028.

[7] Cao S.G., Rees N.W., Feng G., *Analysis and Design for a Class of Complex Control System*, part II: *Fuzzy Controller Design*, „Automatica" 1997, vol. 33, No. 6, s. 1029–1039.

[8] Chmielowski W., Twaróg B., *Regulator rozmyty sterujący przejściem fali powodziowej przez zbiornik retencyjny.* „Czasopismo Techniczne PK", z16Ś/2006.

[9] Hajek M., *Optimization of Fuzzy Rules by Using a Genetic Algorithm. Proceedings of the Third International Conference on Automation*, „Robotics and Computer Vision" 1994, vol. 4, s. 2111–2115.

[10] Kacprzyk J., *Wieloetapowe sterowanie rozmyte*, Warszawa 2001.

[11] Kageyama Sh., *Blood Glucose Control by a Fuzzy Control System*, *Proceedings of the First International Conference on Fuzzy Logic and Neural Networks*, Iizuka 1990.

[12] Łachwa A., *Rozmyty świat zbiorów, liczb, relacji, faktów, reguł i decyzji*, Warszawa 2001.

[13] Piegat A., *Modelowanie i sterowanie rozmyte*, Warszawa 2003.

[14] Rutkowska D., Piliński M., Rutkowski L., *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*, Warszawa-Łódź 1997.

[15] Tobi T., Hanafusa T., *A Practical Application of Fuzzy Control for an Air-conditioning System*, „International Journal of Approximate Reasoning" 1991, No. 5, s. 331–348.

[16] Yager R.R., Filev D.P., *Podstawy modelowania i sterowania rozmytego*, Warszawa 1995.

Izabela Godyń
Wojciech Z. Chmielowski

Zastosowanie rozmytych modeli Takagi–Sugeno do prognozowania poborów wody w gospodarce

1. Wstęp

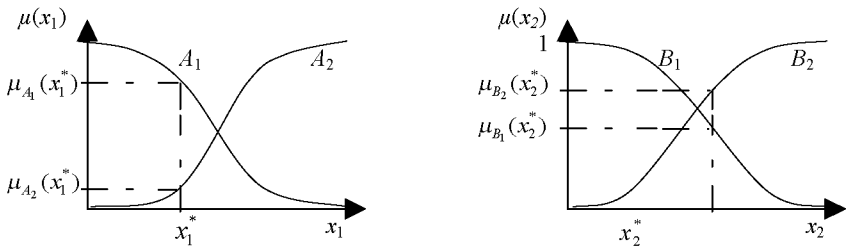
Zmiany w wodochłonności produkcji gospodarki są w ostatnich latach decydującym czynnikiem wpływającym na dynamikę poborów wody. W artykule zaprezentowano próbę zamodelowania zmienności wodochłonności przy użyciu modeli rozmytych o architekturze Takagi–Sugeno. Następnie na podstawie oszacowanych modeli postawiono prognozę zapotrzebowania gospodarki na wodę.

2. Wnioskowanie rozmyte – model Takagi–Sugeno

Model Takagi–Sugeno jest drugim obok modelu Mamdaniego najpowszechniej używanym modelem opartym na wnioskowaniu rozmytym. Jego działanie opiera się na wyciąganiu wniosków o wartościach zmiennej wyjściowej na podstawie wartości zmiennych wejściowych oraz relacji $wy = f(we)$ stworzonych według zasad wnioskowania rozmytego. Model działa w trzech blokach: rozmywania, wnioskowania i wyznaczania ostrej wartości wyjściowej. Poniżej zostaną przedstawione matematyczny zapis i działanie przykładowego modelu rozmytego o dwóch zmiennych wejściowych i jednej zmiennej wyjściowej.

Każda ze zmiennych wejściowych (x_1 i x_2) zdefiniowana jest przez dwie wartości lingwistyczne, np. *mały* i *duży*, oraz odpowiadające im dwa zbiory rozmyte: A_1 (*mały* x_1) i A_2 (*duży* x_1) oraz B_1(*mały* x_2) i B_2 (*duży* x_2). Rozmywanie wartości ostrych odbywa się przez odniesienie ich do zdefiniowanych zbiorów rozmy-

tych reprezentujących poszczególne wartości lingwistyczne. W wyniku tego procesu otrzymuje się wartości stopni przynależności $\mu_{A_i}(x_1^*)$, $\mu_{B_i}(x_2^*)$ wartości ostrych x_1^*, x_2^* do poszczególnych zbiorów rozmytych A_1, A_2, B_1 i B_2, czyli określenie, w jakim stopniu x_1^*, x_2^* są *małe* i *duże*.



Rysunek 1. Przykładowe zdefiniowanie zmiennych w postaci zbiorów rozmytych oraz rozmycie wartości ostrych x_1^*, x_2^*

Zbiory rozmyte obrazujące poszczególne wartości lingwistyczne mogą być opisane przez wiele rodzajów funkcji przynależności, np. funkcję typu gaussowskiego, która w programie MATLAB określona jest wzorem:

$$f(x,\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}} \tag{1}$$

gdzie: σ i c to parametry funkcji przynależności.

Drugim blokiem modelu rozmytego jest blok wnioskowania. Jest to podstawowa część modelu – przetwarza dane wejściowe na odpowiednią wartość wyjścia. Wnioskowanie odbywa się na podstawie bazy reguł, które opisują relacje między zmiennymi wejściowymi i wyjściowymi (w kategoriach wartości lingwistycznych). Wielkość wyjściowa (konkluzja reguły) w modelu Takagi–Sugeno jest zapisana w formie funkcyjnej zależności f między wejściami i wyjściami. Dla przyjętej struktury modelu baza reguł może mieć postać:

R_1: jeżeli x_1 jest *mały* oraz x_2 jest *mały*, to $y = f_1(x_1, x_2)$
R_2: jeżeli x_1 jest *mały* oraz x_2 jest *duży*, to $y = f_2(x_1, x_2)$
R_3: jeżeli x_1 jest *duży* oraz x_2 jest *mały*, to $y = f_3(x_1, x_2)$
R_4: jeżeli x_1 jest *duży* oraz x_2 jest *duży*, to $y = f_4(x_1, x_2)$

lub:

R_1: jeżeli $(x_1 = A_1)$ oraz $(x_2 = B_1)$, to $y_1 = f_1(x_1, x_2)$
R_2: jeżeli $(x_1 = A_1)$ oraz $(x_2 = B_2)$, to $y_2 = f_2(x_1, x_2)$
R_3: jeżeli $(x_1 = A_2)$ oraz $(x_2 = B_1)$, to $y_3 = f_3(x_1, x_2)$
R_4: jeżeli $(x_1 = A_2)$ oraz $(x_2 = B_2)$, to $y_4 = f_4(x_1, x_2)$

Wnioskowanie na podstawie bazy reguł odbywa się przez określenie stopnia aktywacji (w_k) przesłanek reguł, wyznaczenie konkluzji poszczególnych aktywowanych reguł poprzez obliczenie wartości wyjściowych z odpowiednich zależności funkcyjnych $y_k = f_k(x_1, x_2)$.

Stopień aktywacji reguły (w_k) jest wyznaczany jako stopień spełnienia przesłanki reguły. Przy konkluzji dwóch przesłanek prostych stopień spełnienia całej przesłanki może być wyznaczony jako iloczyn dwóch zbiorów rozmytych za pomocą operatora, np. MIN lub PROD:

$$w_k = \mu_{A_i \cap B_j}\left(x_1^*, x_2^*\right) = \min\left(\mu_{A_i}\left(x_1^*\right), \mu_{B_j}\left(x_2^*\right)\right) \tag{2}$$

lub

$$w_k = \mu_{A_i \cap B_j}\left(x_1^*, x_2^*\right) = \mu_{A_i}\left(x_1^*\right) \cdot \mu_{B_j}\left(x_2^*\right) \tag{3}$$

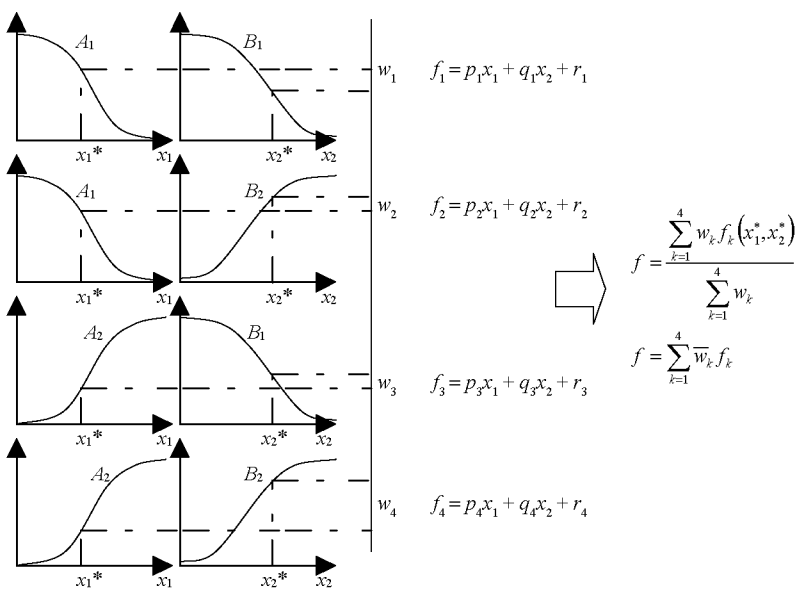
W ostatnim bloku modelu jest obliczana wartość wynikowa y. Jest ona wyznaczana jako średnia ważona z wartości otrzymywanych z aktywowanych reguł, a wagami są stopnie aktywacji konkluzji reguł:

$$y = \frac{\sum_{k=1}^{4} w_k f_k\left(x_1^*, x_2^*\right)}{\sum_{k=1}^{4} w_k} = \frac{\sum_{k=1}^{4} \mu_{A_i \cap B_j}\left(x_1^*, x_2^*\right) f_k\left(x_1^*, x_2^*\right)}{\sum_{k=1}^{4} \mu_{A_i \cap B_j}\left(x_1^*, x_2^*\right)} \tag{4}$$

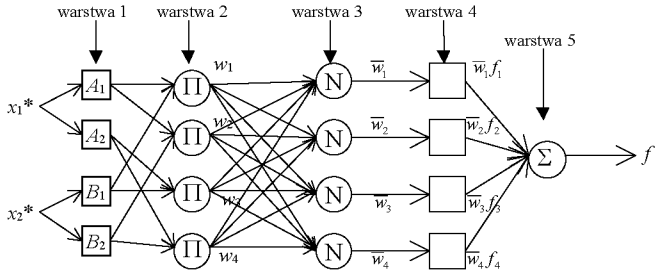
3. ANFIS

Wymagane w konkluzjach reguł zależności funkcyjne dotyczące analizowanych zmiennych bardzo często nie są znane, stąd pierwszym krokiem jest ich estymacja. Środowisko MATLAB firmy The MathWorks, Inc. w pakiecie Simulink zawiera przybornik Fuzzy Logic Toolbox, umożliwiający projektowanie oraz analizę (w tym graficzną) modeli rozmytych. Wśród wielu gotowych funkcji oraz narzędzi opracowano również zaawansowane narzędzie ANFIS (Adaptive-Network-Based Fuzzy Inference System), które pozwala na zbudowanie modelu rozmytego o parametrach dobieranych przez sieć neuronową. Koncepcję konstruowania takiego modelu stworzył w 1993 roku Jang [3]. Na rysunku 2 przedstawiono przykładowy model Takagi–Sugeno i odpowiadającą mu sieć neurorozmytą ANFIS:

(a)



(b)



Rysunek 2. Schemat modelu Takagi–Sugeno (a)
i odpowiadającej mu adaptacyjnej sieci neurorozmytej (b)
Źródło: [3].

Sieć składa się z pięciu warstw – kwadratami zaznaczono węzły adaptacyjne, w których szacowane są parametry, węzły bez parametrów są oznaczone okręgami. Węzły warstwy 1 zawierają parametry funkcji przynależności zbiorów rozmytych opisujących wartości lingwistyczne zmiennych wejściowych. W warstwie 1 obliczane są stopnie przynależności wartości ostrych do zbiorów rozmytych (np. według wzoru (1)). Warstwa 2 nie ma parametrów, oblicza poziomy

aktywacji poszczególnych reguł według wzoru (2) lub innej T-normy. Warstwa 3 również nie posiada parametrów, oblicza unormowane stopnie aktywacji reguł według wzoru:

$$\overline{w}_k = \frac{w_k}{\sum_{k=1}^{4} w_k} \tag{5}$$

Warstwa 4 to warstwa obliczania konkluzji. Ma parametry funkcji $y = f(x_1, x_2)$, czyli funkcji zależności między zmiennymi wejściowymi i wyjściową. Warstwa 5, ostatnia, wyznacza wartość ostrą – wyjściową z modelu (według wzoru (4)).

Tworzenie modelu Takagi–Sugeno w środowisku MATLAB przy zastosowaniu ANFIS wymaga zdefiniowania wstępnej struktury modelu. Model jest tworzony na podstawie pewnych zadanych wielkości – należy podać rodzaj i liczbę funkcji przynależności (liczbę wartości lingwistycznych), a także przyjąć rodzaj zależności funkcyjnej (dopuszczalne są dwa rozwiązania: zależność liniowa lub założenie, że wyjście jest stałe, ale różne dla poszczególnych przedziałów uniwersum). Następnie uruchamiana jest adaptacyjna sieć neuronowa, która bazując na danych historycznych, uczy się i dobiera parametry modelu (model poddawany jest strojeniu: zmiana parametrów funkcji przynależności zmiennych wejściowych, tworzenie bazy reguł, a także obliczanie parametrów funkcji zależności). Nauka sieci może być prowadzona algorytmem wstecznej propagacji błędu z metodą największego spadku lub metodą hybrydową. Metoda hybrydowa to jednoczesne zastosowanie dwóch metod – metody najmniejszych kwadratów oraz metody wstecznej propagacji błędu z gradientem. Metoda najmniejszych kwadratów jest używana w ramach *forward pass* (obliczeń do przodu), szacuje parametry warstwy konkluzji – parametry funkcji $wy = f(we)$. Natomiast metoda gradientowa jest używana w fazie *backward pass* (propagacji błędu) i dobiera parametry warstwy przesłanek, parametry funkcji przynależności.

4. Modelowanie zmienności wodochłonności sektorów gospodarki

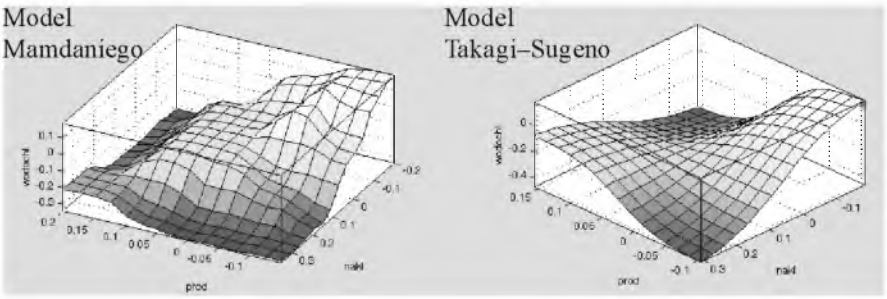
Możliwości zastosowania sieci neurorozmytej ANFIS do modelowania zmian współczynnika wodochłonności są dość ograniczone ze względu na szczupłość zbioru danych historycznych (tylko 12 lat). A model Takagi–Sugeno zawiera znaczącą liczbę parametrów wymagających oszacowania. Na ogólną liczbę parametrów składają się parametry funkcji przynależności zmiennych wejściowych oraz parametry funkcji opisujących zmienną wyjściową. Przy przyjęciu do modelowania dwóch zmiennych wejściowych (dynamiki produkcji globalnej oraz dy-

namiki nakładów inwestycyjnych) model o minimalnej liczbie parametrów byłby następujący:

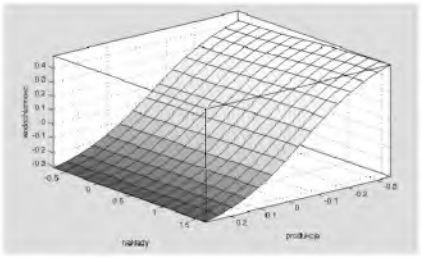
- zmienne wejściowe definiowane przez dwie wartości lingwistyczne o gaussowskich funkcjach przynależności (które mają tylko dwa parametry): liczba parametrów = liczba zmiennych * liczba wartości * liczba parametrów funkcji, czyli $2 \cdot 2 \cdot 2 = 8$;
- zmienna wyjściowa opisana za pomocą stałych wartości dla poszczególnych konkluzji reguł, czyli dla poszczególnych możliwych kombinacji zmiennych wejściowych:
  liczba parametrów = liczba reguł, czyli 4.

Podsumowując, można powiedzieć, że jedna z prostszych struktur modelu – model o dwóch zmiennych wejściowych opisanych przez dwie wartości lingwistyczne – wymaga dobrania 12 parametrów. Ilość danych historycznych (12 lat) jest właściwie na granicy stosowalności tego modelu. Zdecydowano się jednak zbudować modele typu Takagi–Sugeno, aby pokazać, jak efektywnym narzędziem jest estymacja parametrów modelu za pomocą sieci neuronowych. Dla poszczególnych sektorów zbudowano modele dynamiki wodochłonności wód powierzchniowych i podziemnych o wyżej wymienionej strukturze. Podczas gdy budowa modeli Mamdaniego opartych na tych samych danych, a następnie ich strojenie (czyli estymacja parametrów) są bardzo praco- i czasochłonne, estymacja parametrów modeli Sugeno (o wyżej wymienionej strukturze i na takim zbiorze danych historycznych) jest procesem wykonywanym przez sieć w przeciągu kilkunastu sekund.

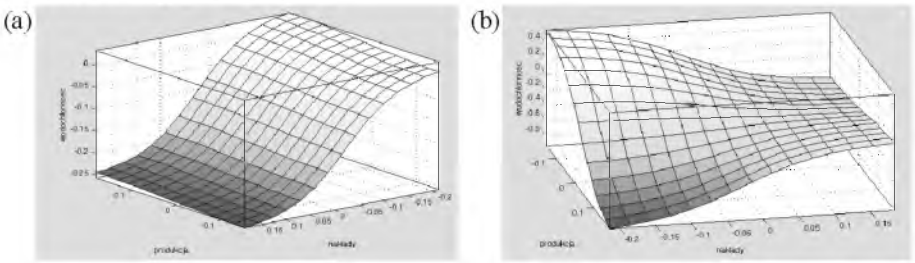
Ostatecznie przyjęto, że modele zmienności wodochłonności typu Takagi–Sugeno będą się opierały, tak jak modele Mamdaniego, na dwóch zmiennych wejściowych: dynamice produkcji globalnej i dynamice nakładów inwestycyjnych. Obydwie zmienne wejściowe zdecydowano się opisywać dwoma wartościami lingwistycznymi o funkcjach przynależności typu gaussowskiego. Zmienna wyjściowa, a dokładniej funkcje zależności pomiędzy wejściami i wyjściem zostały przyjęte jako stałe dla poszczególnych reguł (czyli dla poszczególnych przedziałów zmienności wejść). Dla tak przyjętych założeń oszacowano modele dla poszczególnych sektorów gospodarki. Przyjęto podział gospodarki na 7 sektorów: A (rolnictwo), B (rybactwo), C (górnictwo), D (przetwórstwo przemysłowe), E-en. (energetyka), E-pob. (pobór wód), F–O (pozostałe) – podział dokładnie opisany w artykule [2]. Modelowano zmienność współczynników wodochłonności wód powierzchniowych i podziemnych. Otrzymane modele zależności zmienności współczynników wodochłonności od dynamiki produkcji i nakładów inwestycyjnych są bardzo zbliżone do opracowanych modeli Mamdaniego (opisanych w artykule [2]).



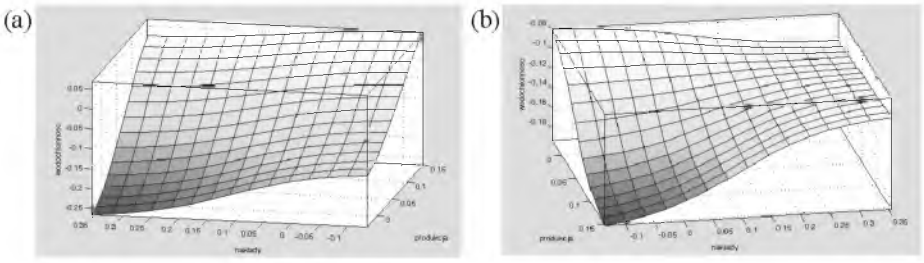
Rysunek 3. Porównanie modeli Mamdaniego i Takagi–Sugeno dynamiki wodochłonności wód powierzchniowych dla sektora A



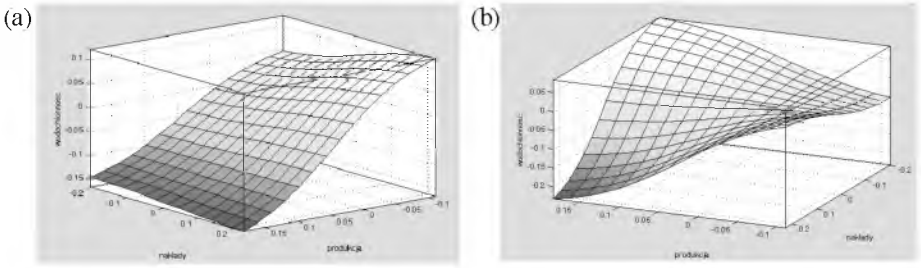
Rysunek 4. Model Takagi–Sugeno dynamiki wodochłonności wód powierzchniowych w sektorze B



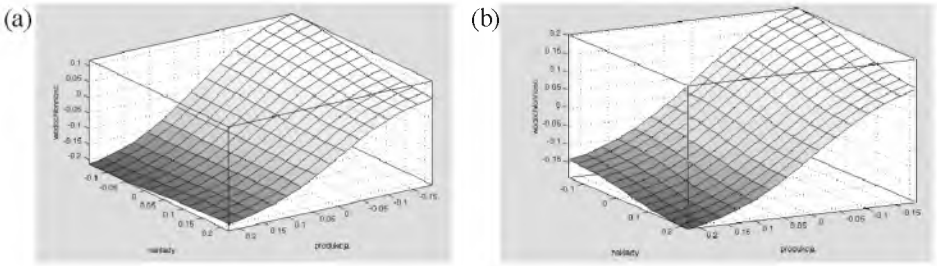
Rysunek 5. Modele Takagi–Sugeno dynamiki wodochłonności wód powierzchniowych (a) i podziemnych (b) w sektorze C



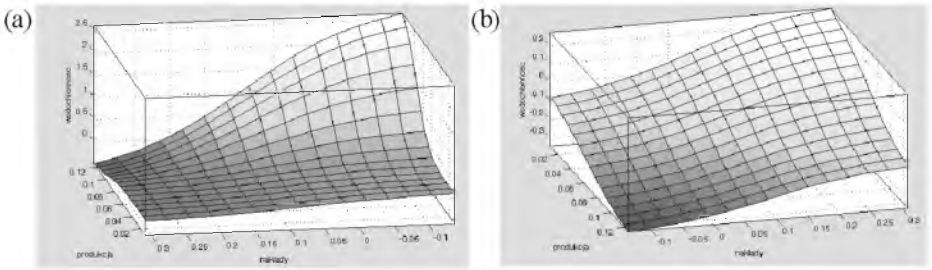
Rysunek 6. Modele Takagi–Sugeno dynamiki wodochłonności wód powierzchniowych (a) i podziemnych (b) w sektorze D



Rysunek 7. Modele Takagi–Sugeno dynamiki wodochłonności wód powierzchniowych (a) i podziemnych (b) w sektorze E-en.



Rysunek 8. Modele Takagi–Sugeno dynamiki wodochłonności wód powierzchniowych (a) i podziemnych (b) w sektorze E-pob.



Rysunek 9. Modele Takagi–Sugeno dynamiki wodochłonności wód powierzchniowych (a) i podziemnych (b) w sektorach F–O

5. Prognoza zużycia wód powierzchniowych i podziemnych przy wykorzystaniu sektorowych modeli dynamiki wodochłonności Takagi–Sugeno

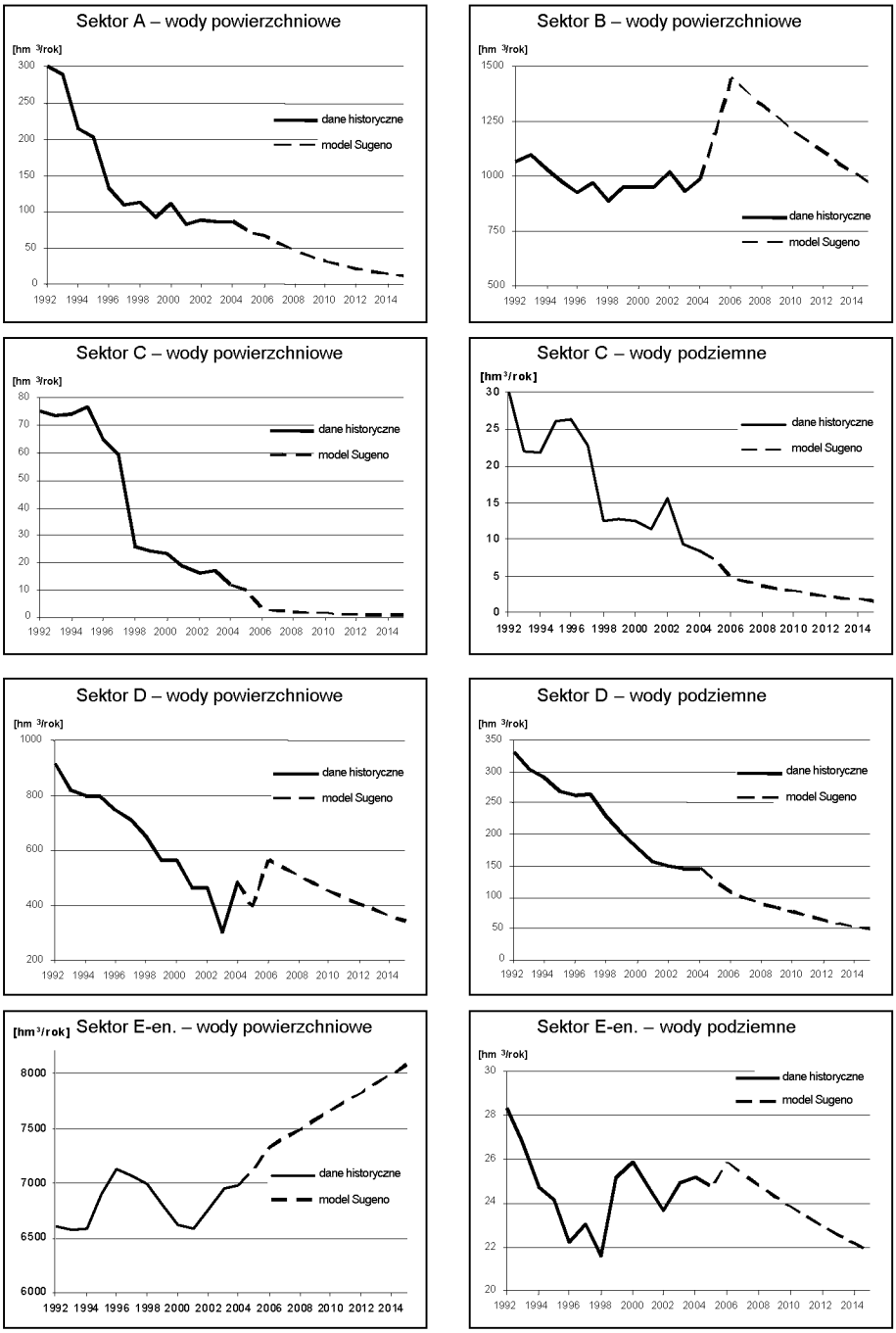
Opracowane modele dynamiki wodochłonności zostały następnie połączone z modelem *input-output* w celu wyznaczenia prognozy zużycia zasobów wodnych. Wejściowymi wielkościami do modelu były tak jak poprzednio: macierz powiązań międzysektorowych, prognoza dynamiki wartości dodanej oraz prognoza dynamiki nakładów inwestycyjnych. Prognozę wodochłonności gospodarki przygotowaną przy użyciu tego modelu przedstawiono w tabelach poniżej.

Tabela 1. Prognozowane współczynniki wodochłonności wód powierzchniowych i podziemnych

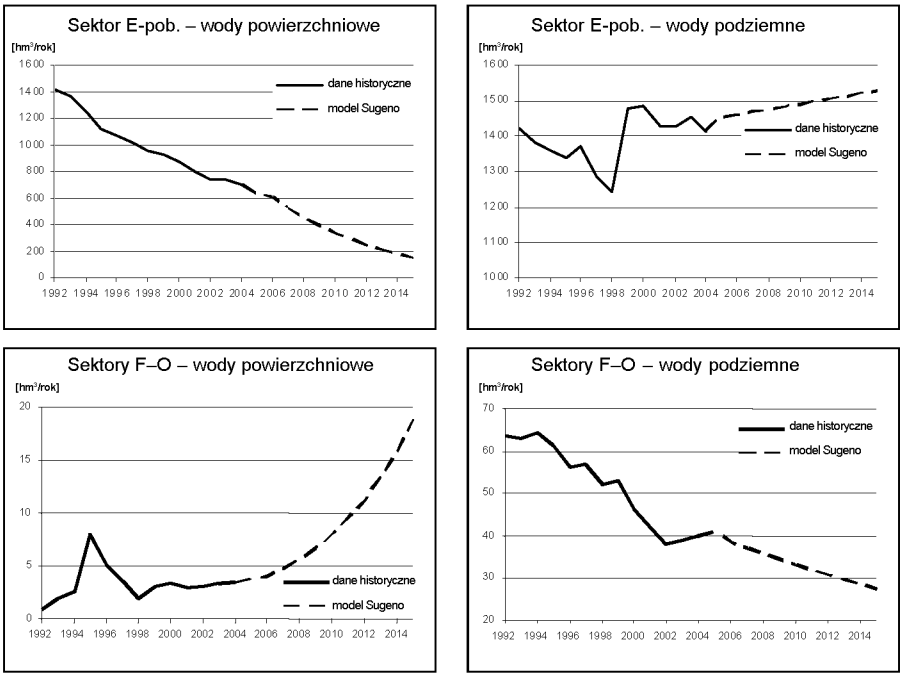
Lp.	Sektor		Dane historyczne	Prognoza współczynników zużycia wód powierzchniowych [m³/tys. zł]			2015/2004
			2004	2007	2010	2015	
1.	Rolnictwo	A	1,1919	0,7279	0,3773	0,1244	10%
2.	Rybactwo	B	2 494,1772	2 622,9628	2 020,3638	1 297,5020	52%
3.	Górnictwo	C	0,3530	0,0710	0,0444	0,0203	6%
4.	Przetwórstwo przemysłowe	D	0,8118	0,8547	0,6303	0,3815	47%
5.	Energetyka	E-en.	110,4739	113,6651	113,1826	110,7269	100%
6.	Pobór wód	E-pob.	100,0683	71,6878	44,4300	19,5959	20%
7.	Pozostałe	F–O	0,0033	0,0042	0,0061	0,0114	341%
Lp.	**Sektor**		**Dane historyczne**	**Prognoza współczynników zużycia wód podziemnych [m³/tys. zł]**			**2015/2004**
			2004	**2007**	**2010**	**2015**	
1.	Rolnictwo	A	0,0000	0,0000	0,0000	0,0000	–
2.	Rybactwo	B	0,0000	0,0000	0,0000	0,0000	–
3.	Górnictwo	C	0,2483	0,1302	0,0906	0,0471	19%
4.	Przetwórstwo przemysłowe	D	0,2438	0,1574	0,1061	0,0548	22%
5.	Energetyka	E-en.	0,3994	0,3882	0,3522	0,2987	75%
6.	Pobór wód	E-pob.	200,7053	200,2881	197,3247	190,5010	95%
7.	Pozostałe	F–O	0,0385	0,0326	0,0254	0,0166	43%

Tabela 2. Prognozowany pobór wód powierzchniowych i podziemnych

Lp.	Sektor		Dane historyczne	Prognozowany pobór wód powierzchniowych [hm³/rok]			2015/ 2004
			2004	2007	2010	2015	
1.	Rolnictwo	A	86,20	55,49	31,24	11,92	14%
2.	Rybactwo	B	985,20	1 383,52	1 211,29	967,28	98%
3.	Górnictwo	C	11,80	2,30	1,44	0,68	6%
4.	Przetwórstwo przemysłowe	D	485,50	533,51	450,50	342,85	71%
5.	Energetyka	E-en.	6 971,10	7 406,67	7 655,42	8 077,48	116%
6.	Pobór wód	E-pob.	703,70	525,81	335,90	157,18	22%
7.	Pozostałe	F–O	3,46	4,80	8,00	18,84	545%
OGÓŁEM			9 246,96	9 912,10	9 693,79	9 576,22	104%
Lp.	**Sektor**		**Dane historyczne**	**Prognozowany pobór wód podziemnych [hm³/rok]**			**2015/ 2004**
			2004	**2007**	**2010**	**2015**	
1.	Rolnictwo	A	0,00	0,00	0,00	0,00	–
2.	Rybactwo	B	0,00	0,00	0,00	0,00	–
3.	Górnictwo	C	8,30	4,21	2,94	1,57	19%
4.	Przetwórstwo przemysłowe	D	145,80	98,27	75,82	49,25	34%
5.	Energetyka	E-en.	25,20	25,29	23,82	21,79	86%
6.	Pobór wód	E-pob.	1 411,40	1 469,06	1 491,82	1 528,04	108%
7.	Pozostałe	F–O	39,91	37,17	33,25	27,46	69%
OGÓŁEM			1 630,61	1 634,01	1 627,65	1 628,11	100%



Rysunek 10. Prognoza poborów wód powierzchniowych i podziemnych w sektorach: A, B, C, D i E-en.



Rysunek 11. Prognoza poborów wód powierzchniowych i podziemnych w sektorze E-pob. i w grupie pozostałych sektorów F–O

Otrzymana prognoza poborów wód powierzchniowych i podziemnych w sektorach oparta na modelach zmienności wodochłonności typu Takagi–Sugeno to: wzrost poborów z wód powierzchniowych o około 4% i prawie stały poziom poborów z wód podziemnych. Są to wartości całkowitego poboru, realizowanego przez całą gospodarkę. Zmiany dotyczące poborów poszczególnych sektorów są dużo bardziej zróżnicowane. Największe zmiany w zakresie korzystania z wód powierzchniowych modele prognozują w sektorach A, C i E-pob. – przewidywane są bardzo wysokie spadki współczynników wodochłonności, powodujące wysokie spadki w poborach – o 80–90%. Istotne zmiany w odwrotnym kierunku (wzrost współczynnika wodochłonności) prognozowane są w grupie sektorów F–O i w związku z tym przewiduje się również wysoki wzrost poboru w tych sektorach. Natomiast wodochłonność wód podziemnych obniży się we wszystkich sektorach: znacząco w sektorach C i D – o około 80%, w sektorach F–O o około 55%, w E-en. – 25%; w sektorze E-pob. spadki współczynników wodochłonności będą rzędu 5%. Takie zmiany w wodochłonności spowodują, przy wzroście produkcji, wzrost poborów w sektorze E-pob. o 8%, niskie spadki w sektorze E-en. (o 14%). W pozostałych sektorach pobory obniżą się znacznie – o 30% w sektorze F–O, o 65% w sektorze D, a w sektorze C – o ponad 80%.

6. Porównanie prognoz zużycia zasobów wodnych uzyskanych w modelach Mamdaniego i Takagi–Sugeno

Zmienność wodochłonności poszczególnych sektorów gospodarki została zamodelowana przy użyciu technik modelowania rozmytego według architektury Mamdaniego oraz Takagi–Sugeno. W celu porównania jakości modeli obliczono na podstawie danych historycznych i wartości otrzymanych z modeli:

• średni błąd kwadratowy:

$$\delta = \frac{\left(b - \tilde{b}\right)^2}{n} \tag{6}$$

gdzie: b – wartości historyczne współczynnika wodochłonności,
$\tilde{b}$ – wartości współczynnika wodochłonności otrzymane z modelu,
n – liczba obserwacji,

• średni błąd względny:

$$s = \frac{\sqrt{\delta}}{\bar{b}} \tag{7}$$

gdzie: $\bar{b}$ – średni współczynnik wodochłonności (z danych historycznych).

Tabela 3. Średni błąd kwadratowy i błąd względny dla modeli Mamdaniego i Takagi–Sugeno: (a) współczynniki wodochłonności wód powierzchniowych; (b) współczynniki wodochłonności wód podziemnych

(a)

Lp.	Sektor		Średni błąd kwadratowy		Średni błąd względny	
			modele Mamdaniego	modele Takagi–Sugeno	modele Mamdaniego	modele Takagi–Sugeno
1.	Rolnictwo	A	0,025	0,031	10%	11%
2.	Rybactwo	B	26186	33175	12%	13%
3.	Górnictwo	C	0,032	0,039	15%	17%
4.	Przetwórstwo przemysłowe	D	0,028	0,029	12%	12%
5.	Energetyka	E-en.	25	5	4%	2%
6.	Pobór wód	E-pob.	523	993	12%	16%
7.	Pozostałe	F–O	0,000004	0,000010	41%	66%

(b)

Lp.	Sektor		Średni błąd kwadratowy		Średni błąd względny	
			modele Mamdaniego	modele Takagi–Sugeno	modele Mamdaniego	modele Takagi–Sugeno
1.	Górnictwo	C	0,0108	0,0112	21%	21%
2.	Przetwórstwo przemysłowe	D	0,0010	0,0013	6%	7%
3.	Energetyka	E-en.	0,00088	0,00046	6%	4%
4.	Pobór wód	E-pob.	340	428	7%	8%
5.	Pozostałe	F–O	0,0000234	0,0000061	7%	3%

Biorąc pod uwagę wartości obliczonych błędów, modele Mamdaniego i Takagi–Sugeno dają porównywalne wyniki, choć dla większości przypadków lepsze dopasowanie osiągnięto w modelach Mamdaniego.

Wyżej wymienione modele zmienności wodochłonności dały znacznie różniące się prognozy poborów wód powierzchniowych i podziemnych. W celu porównania otrzymywanych wartości prognoz poborów wykonano także symulację zapotrzebowania na wodę gospodarki przy założeniu, że współczynniki wodochłonności utrzymają się w całym okresie prognozy na poziomie z 2004 roku.

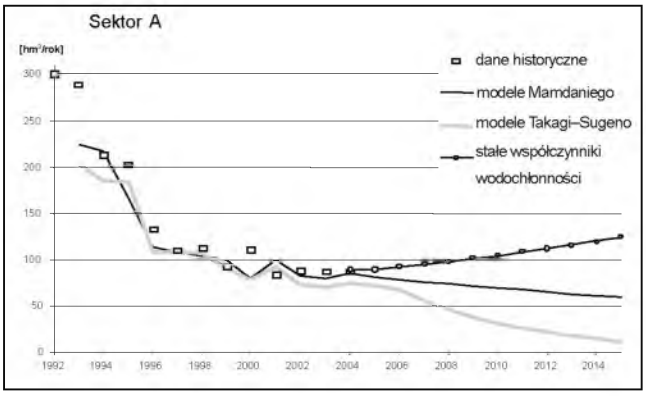
Tabela 4. Porównanie prognoz poborów wód w 2015 roku [hm^3/rok] otrzymanych z modeli: (a) pobór wód powierzchniowych; (b) pobór wód podziemnych

(a)

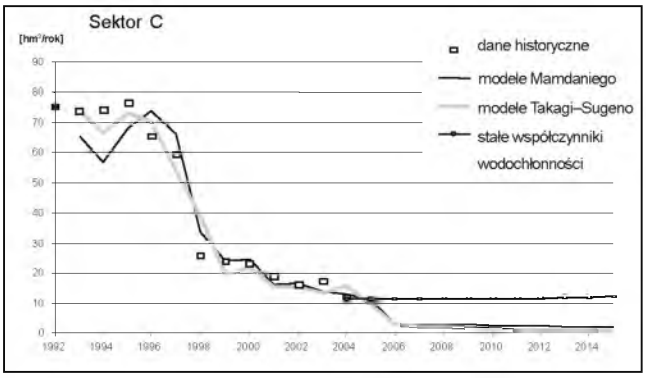
Lp.	Sektor		Stałe współczynniki 2004 rok	Modele Mamdaniego	Modele Takagi–Sugeno
1.	Rolnictwo	A	124,32	59,03	11,92
2.	Rybactwo	B	1 969,83	1 885,49	967,28
3.	Górnictwo	C	12,05	2,14	0,68
4.	Przetwórstwo przemysłowe	D	771,10	225,03	342,85
5.	Energetyka	E-en.	10 669,30	7 607,97	8 077,48
6.	Pobór wód	E-pob.	889,08	447,95	157,18
7.	Pozostałe	F–O	5,83	6,47	18,84
	OGÓŁEM		14 441,49	10 234,09	9 576,22

(b)

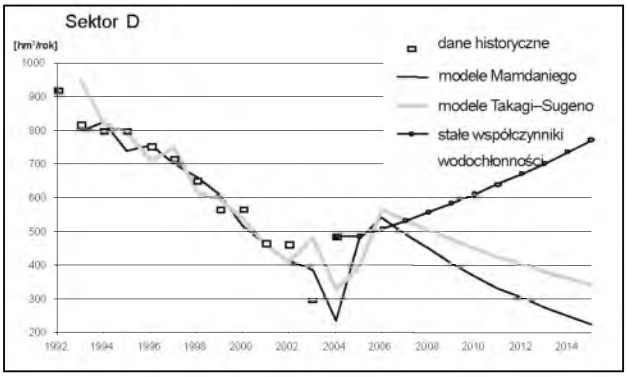
Lp.	Sektor		Stałe współczynniki 2004 rok	Modele Mamdaniego	Modele Takagi–Sugeno
1.	Górnictwo	C	8,48	4,18	1,57
2.	Przetwórstwo przemysłowe	D	231,57	56,15	49,25
3.	Energetyka	E-en.	35,06	23,95	21,79
4.	Pobór wód	E-pob.	1 783,21	1 412,39	1 528,04
5.	Pozostałe	F–O	67,28	18,82	27,46
	OGÓŁEM		2 125,59	1 515,49	1 628,11



Rysunek 12. Porównanie wyników prognozowania poborów wód powierzchniowych w sektorze A



Rysunek 13. Porównanie wyników prognozowania poborów wód powierzchniowych w sektorze C



Rysunek 14. Porównanie wyników prognozowania poborów wód powierzchniowych w sektorze D

Przy stałych (na poziomie z 2004 roku) współczynnikach wodochłonności zmiana w poborach jest zależna jedynie od zmiany poziomu produkcji, a ponieważ ogólnie dla całej gospodarki prognozowany jest wzrost produkcji globalnej, to pobory także wzrosną. Wzrost poboru nie będzie jednak tak wysoki jak wzrost ogólnej produkcji całej gospodarki (corocznie 5%, czyli w ciągu 11 lat wzrośnie o około 60–70%), ponieważ rozwój gospodarczy nie jest taki sam dla wszystkich sektorów, np. dla sektora C przewidywany jest spadek produkcji (a tym samym spadek poboru), a dla sektora E (najbardziej wodochłonnego w całej gospodarce) prognozuje się 3% dynamikę produkcji na te lata. Największe różnice pomiędzy wynikami symulacji wodochłonności gospodarki według stałych współczynników z 2004 roku i współczynników zużycia wody są widoczne w poborach wód powierzchniowych i podziemnych w sektorach D, E-en. i E-pob. Współczynniki wodochłonności wód powierzchniowych i podziemnych tych sektorów mają kluczowy wpływ na prognozowane wartości poborów. Dynamika współczynników zużycia wód powierzchniowych i podziemnych w wyżej wymienionych sektorach jest prognozowana jako malejąca, w związku z czym nastąpi w nich znaczne zmniejszenie poboru w 2015 roku. Wartości sumarycznych poborów wód powierzchniowych i podziemnych w 2015 roku wyniosą: według modelu o stałych współczynnikach – 16 500 hm^3/rok, według modeli Mamdaniego – 11 700, a według modeli Takagi–Sugeno – 11 100 hm^3/rok. Wyniki te nie są alarmujące, w latach osiemdziesiątych ubiegłego wieku pobory wynosiły około 15 500 hm^3 rocznie. Dlatego na podstawie otrzymanych wartości poborów i ich porównania z maksymalnymi wartościami historycznymi można stwierdzić, że zasoby wodne powinny być wystarczające do pokrycia prognozowanego zapotrzebowania na wodę. Trzeba jednak podkreślić, że prognoza ma zasięg obszarowy kraju. Na poziomie bardziej szczegółowym – w poszczególnych regionach wodnych czy zlewniach – mogą

wystąpić tendencje odmienne od wartości średnich krajowych i tam rozwój gospodarczy może się spotkać z barierą niewystarczających zasobów wodnych i/lub infrastruktury technicznej.

7. Wnioski

Tworzenie modeli rozmytych o architekturze Takagi–Sugeno wymaga zdefiniowania matematycznego funkcji opisujących zależność pomiędzy zmiennymi wejściowymi i wyjściową $wy = f(we)$. Bardzo często funkcje te nie są znane bądź ich oszacowanie jest praco- i czasochłonne. Tak jest w przypadku zmienności wodochłonności gospodarki. Nieznajomość zależności funkcyjnych nie wyklucza możliwości zastosowania tego typu modelu. Modele przedstawione w tym opracowaniu zostały zbudowane za pomocą adaptacyjnych sieci neuronowych, które oszacowały parametry modeli Takagi–Sugeno. Wykorzystano adaptacyjną sieć neuronową ANFIS – gotowe narzędzie dostępne w programie MATLAB w przyborniku Fuzzy Logic Toolbox.

Bibliografia

[1] *Fuzzy Logic Toolbox User's Guide*, The MathWorks, Inc., 1995–2007, http://www.mathworks.com/access/helpdesk/help/pdf_doc/fuzzy/fuzzy.pdf.
[2] Godyń I., Chmielowski W.Z., *Zastosowanie wnioskowania rozmytego do prognozowania zmienności wodochłonności i zużycia wody w gospodarce* [niniejszy zeszyt].
[3] Jang J.-S.R., ANFIS. *Adaptive-Network-Based Fuzzy Inference System*, „IEEE Transactions on Systems, Man, and Cybernetics" 1993, vol. 23, No. 3.
[4] Łachwa A., *Rozmyty świat zbiorów, relacji i reguł*, Warszawa 2001.
[5] Piegat A., *Modelowanie i sterowanie rozmyte*, Warszawa 1999.

Izabela Godyń
Wojciech Z. Chmielowski

Zastosowanie wnioskowania rozmytego do prognozowania zmienności wodochłonności i zużycia wody w gospodarce

1. Wstęp

Prognozowanie zapotrzebowania na wodę gospodarki jest istotnym elementem wspomagania zarządzania zasobami wodnymi, a w szczególności planowania zadań inwestycyjnych dotyczących rozwoju infrastruktury. Wielkością decydującą o poziomie zapotrzebowania na wodę, obok rozwoju gospodarczego, jest wodochłonność produkcji. W ostatnich latach można obserwować spadek poborów towarzyszący wzrostowi produkcji w większości sektorów gospodarki. Niniejszy artykuł przedstawia próbę modelowania zmienności wodochłonności przy wykorzystaniu wnioskowania rozmytego, a następnie połączenia modelu *input-output* z rozmytymi modelami zmienności wodochłonności poszczególnych gałęzi gospodarki w celu budowy prognoz poborów wody przez gospodarkę.

2. Rozbudowany model *input-output*

Podstawy teoretyczne modelowania *input-output* (analizy przepływów międzygałęziowych) stworzone zostały przez W. Leontiefa w 1941 roku. Analizy te opierają się na tablicach zależności pomiędzy sektorami gospodarki. Klasyczny model *input-output* Leontiefa jest równaniem w zapisie macierzowym:

$$\mathbf{x} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{y} \tag{1}$$

gdzie:

- $\mathbf{x}$ – wektor $(n \times 1)$ produkcji globalnej we wszystkich gałęziach,
- $\mathbf{I}$ – macierz jednostkowa,
- $\mathbf{A}$ – macierz $(n \times n)$ współczynników produktochłonności wyznaczanych jako:

$$a_{ij} = \frac{x_{ij}}{x_j},$$

gdzie:

x_i – produkcja globalna w gałęzi *i*,
x_{ij} – zużycie pośrednie produktów gałęzi *i* przez gałąź *j*,
y – wektor $(n \times 1)$ zużycia końcowego we wszystkich gałęziach.

Pewną modyfikację (a właściwie alternatywę) modelu Leontiefa zaproponował w 1964 roku Ghosh, który również wykorzystał tablicę *input-output*, jednak do wyznaczania produkcji globalnej posłużył się wartością dodaną (a nie jak Leontief zużyciem końcowym). Model Ghosha ma postać:

$$\mathbf{x}^T = \left(\mathbf{v}(\mathbf{I} - \mathbf{G})^{-1}\right)^T \tag{2}$$

gdzie:

x – wektor $(n \times 1)$ produkcji globalnej dla gałęzi,
$()^T$ – transpozycja wektora lub macierzy,
v – wektor $(1 \times n)$ wartości dodanej według gałęzi,
G – macierz $(n \times n)$ współczynników $g_{ij} = \frac{x_{ij}}{x_i}$,
pozostałe oznaczenia jak wyżej.

Dla potrzeb analiz ekonomiczno-środowiskowych rozbudowuje się tablice przepływów międzygałęziowych o dane o zużyciu zasobów naturalnych oraz wytwarzanych zanieczyszczeniach. Modelowanie opiera się na wyznaczeniu współczynników zużycia b_{kj} dóbr środowiskowych w przeliczeniu na jednostkę pieniężną produkcji gałęzi gospodarki *j*.

$$b_{kj} = \frac{Z_{kj}}{x_j} \tag{3}$$

gdzie:

Z_{kj} – zużycie dobra środowiskowego *k* (lub emisja zanieczyszczenia *k*) przez sektor gospodarki *j*,
x_j – produkcja globalna sektora gospodarki *j*.

Globalne zużycie może być zapisane *w zależności od produkcji globalnej*. Korzystając z zapisu równań (1) i (2), można również wyznaczyć *zależność zużycia dóbr środowiskowych od zużycia końcowego produkcji gospodarki* lub *wartości dodanej*:

$$\mathbf{z} = \mathbf{B}\mathbf{x} = \mathbf{B}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{y} = \mathbf{B}\left(\mathbf{v}(\mathbf{I} - \mathbf{G})^{-1}\right)^T \tag{4}$$

gdzie:

B – macierz $(k \times n)$ współczynników zużycia b_{kj} poszczególnych *k* dóbr przez *n* sektorów,

z – wektor $(k \times 1)$ zużycia k dóbr środowiskowych (lub emisji k zanieczyszczeń) przez całą gospodarkę.

Prognozowanie oparte na powiązaniach międzygałęziowych w gospodarce wnosi wiele informacji. Przykład zamieszczony poniżej ilustruje różnice w prognozowaniu wykorzystującym podejście klasyczne i podejście wspomagane modelem *input-output*:

Podejście klasyczne:

Prognozowany jest dynamiczny rozwój sektora X gospodarki. Zużycie wody przez ten sektor jest bardzo niskie, prognoza zużycia na podstawie wzrostu produkcji i wymaganych do produkcji zasobów prowadzi do wniosku, że *rozwój gospodarczy* w tym sektorze *nie będzie miał wpływu na zużycie* wody.

Podejście oparte na tablicach *input-output*:

Na podstawie tablic przepływów międzygałęziowych można wyznaczyć sektory, z których produkcji korzysta sektor X – okazuje się, że są one bardzo wodochłonne. Prognoza wzrostu zapotrzebowania na produkty sektora X jest przekładana na związany z tym wzrost produkcji w pozostałych sektorach, a następnie wyznaczany jest wzrost zużycia wody w tych sektorach. Ostatecznie określa się, że *rozwój gospodarczy* w sektorze X *będzie miał wpływ na zużycie* wody.

Do modelowania i prognozowania zużycia wody zdecydowano się stosować model Ghosha z uwagi na to, że dostępność danych (w rozbiciu na sektory) o wartości dodanej $\mathbf{v}(t)$ jest większa niż danych o popycie końcowym $\mathbf{y}(t)$. W modelowaniu polskiej gospodarki i wielkości jej potrzeb wodnych należy uwzględniać zmiany współczynników zużycia wód powierzchniowych i podziemnych b_{kj}. Taki dynamiczny model można opisać jako:

$$\mathbf{Z}(t) = \mathbf{B}(t)\left(\mathbf{v}(t)(\mathbf{I} - \mathbf{G})^{-1}\right)^{T} \tag{5}$$

Macierz **G**, obrazująca przepływy produktów pomiędzy gałęziami przemysłu, również ulega zmianie w czasie, jednakże zmiany te nie są decydujące dla zużycia wody, dlatego przyjęto, że **G** jest stała w czasie. Wielkości współczynnika wodochłonności proponuje się prognozować przez budowę modeli opartych na wnioskowaniu rozmytym.

Modelowanie zużycia zasobów wodnych dotyczy dwóch najistotniejszych źródeł poborów – wód powierzchniowych i wód podziemnych. Stąd macierz **B** będzie dwuwierszowa (k = 1, 2) i będzie zawierać współczynniki zużycia wód powierzchniowych oraz współczynniki zużycia wód podziemnych przez poszczególne sektory gospodarki.

3. Wnioskowanie rozmyte – model Mamdaniego

Podstawowym elementem wnioskowania rozmytego jest pojęcie zmiennej lingwistycznej (np. „wodochłonność produkcji"), która przyjmuje wartości lingwistyczne, takie jak: „niska", „bardzo niska", „średnia", „wysoka", „bardzo wysoka" itp. Wartościom lingwistycznym są przypisywane odpowiednie zbiory rozmyte, a zależności między zmiennymi lingwistycznymi – rozmyte zdania warunkowe. Przykładowo: jeśli mamy dwie zmienne lingwistyczne L i K, takie, że wartość zmiennej L jest zbiorem rozmytym A określonym w X oraz wartość zmiennej K jest zbiorem rozmytym B określonym w Y, to zależność między L i K, a właściwie między wartościami A i B, można zapisać jako:

$$\text{Jeżeli } L = A, \text{ to } K = B.$$

Wnioskowanie rozmyte oparte na logice zbiorów rozmytych polega na wyciąganiu wniosków na podstawie reguł opartych na wartościach lingwistycznych. Najczęściej stosowana jest architektura Mamdaniego, w której na podstawie wiedzy eksperta tworzy się bazę reguł postaci, przykładowo: dla dwóch zmiennych wejściowych $x1$ i $x2$ oraz zmiennej wyjściowej y, przyjmujących po trzy wartości lingwistyczne („ujemny" – U, „zero" – Z, „dodatni" – D), możliwe jest utworzenie 9 reguł odpowiadających wszystkim kombinacjom zmiennych wejściowych. Reguły te mogą na przykład mieć postać: „Jeżeli $x1$ przyjmuje wartość U i $x2$ przyjmuje wartość U, to wielkość wyjściowa y przyjmie wartość U", co można zapisać jako:

1. Jeżeli $x1 = U$ oraz $x2 = U$, to $y = U$.

i kolejne reguły:

2. Jeżeli $x1 = U$ oraz $x2 = Z$, to $y = U$.
3. Jeżeli $x1 = U$ oraz $x2 = D$, to $y = Z$.
4. Jeżeli $x1 = Z$ oraz $x2 = U$, to $y = U$.
5. Jeżeli $x1 = Z$ oraz $x2 = Z$, to $y = Z$.
6. Jeżeli $x1 = Z$ oraz $x2 = D$, to $y = D$.
7. Jeżeli $x1 = D$ oraz $x2 = U$, to $y = Z$.
8. Jeżeli $x1 = D$ oraz $x2 = Z$, to $y = D$.
9. Jeżeli $x1 = D$ oraz $x2 = D$, to $y = D$.

Praca modelu Mamdaniego przebiega następująco: do modelu wprowadzane są ostre wielkości zmiennych wejściowych, które są zamieniane na odpowiednie zbiory rozmyte (dana wartość ostra może odpowiadać jednemu lub dwóm zbiorom rozmytym, np. wartość –0,001 będzie w pewnym stopniu przynależała do zbioru „ujemny" oraz do zbioru „zero"), stąd ostre wartości dwóch zmiennych mogą uruchomić od jednej do kilku reguł (w zależności od podziału uniwersum). Każda z tych reguł jest spełniona w pewnym stopniu, ponieważ wejścia miały określone stopnie przynależności do odpowiadających im zbiorów rozmytych. Jeżeli przesłanka reguły składa się z dwóch przesłanek dotyczących dwóch wejść

połączonych spójnikiem koniunkcyjnym „oraz", to stopień przynależności do całej reguły liczy się najczęściej jako stopień przynależności do relacji będącej iloczynem dwóch zbiorów rozmytych, a najczęściej stosowanym do obliczeń operatorem są operatory *t*-normy: minimum MIN oraz iloczyn algebraiczny PROD. W wyniku uruchomienia przykładowo 4 reguł otrzymujemy 4 konkluzje i odpowiadające im wielkości wyjścia (o różnych stopniach przynależności). Końcowy zbiór rozmyty jest sumą konkluzji poszczególnych reguł, czyli sumą zbiorów rozmytych będących wyjściami z poszczególnych reguł. Przynależność do takiej relacji obliczana jest najczęściej jako maksimum MAX lub inny operator typu *s*-normy, np. suma logiczna. W rezultacie w bloku wnioskowania otrzymywana jest wartość zmiennej wyjściowej w postaci zbioru rozmytego. Ostatnim blokiem modelu jest blok wyostrzania – defuzyfikacji, który pozwala na przekształcenie wyjściowego zbioru rozmytego do wyjścia w postaci wielkości ostrej. Opracowano wiele metod defuzyfikacji, najpopularniejsze to: metoda środka maksimum, metoda środka ciężkości, metoda środka sum, szeroko opisane w literaturze przedmiotu (m.in. w: [3, 5, 6]).

4. Modelowanie zmienności współczynników wodochłonności przy użyciu rozmytych modeli Mamdaniego

Modelowanie zmienności współczynnika zużycia wody należy przeprowadzić dla każdego z sektorów gospodarki. Z uwagi na to, że większość danych statystycznych dotyczących zmiennych (nakłady inwestycyjne, prognoza rozwoju sektorów gospodarki) jest dostępna na poziomie głównych sektorów gospodarki, dokonano agregacji struktury gospodarki do przedstawionego w tabeli 1 schematu głównych sektorów.

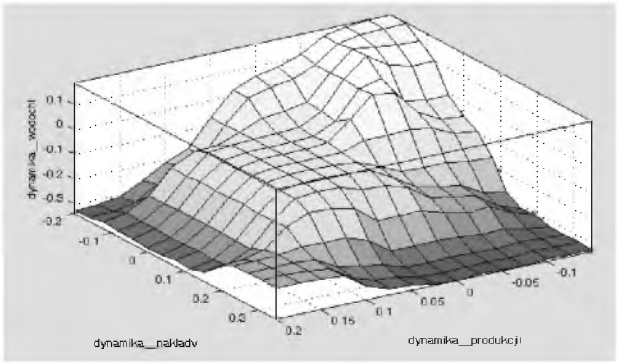
Tabela 1. Schemat gospodarki przyjęty do modelowania

Lp.	Sektor		Udział w poborach w 2004 roku		
			Wody ogółem	Wody powierzchniowe	Wody podziemne
1.	Rolnictwo	A	0,8%	0,9%	–
2.	Rybactwo	B	9,0%	10,6%	–
3.	Górnictwo	C	0,8%	0,1%	0,5%
4.	Przetwórstwo przemysłowe	D	5,9%	5,2%	8,9%
5.	Energetyka	E-en.	63,7%	75,3%	1,5%
6.	Pobór wód	E-pob.	19,2%	7,6%	86,6%
7.	Pozostałe		0,6%	0,0%	2,4%

Na podstawie analizy danych historycznych przyjęto, że do modeli dynamiki wodochłonności (osobno wód podziemnych i powierzchniowych) poszczególnych sektorów gospodarki będą używane dwie zmienne: dynamika produkcji globalnej oraz dynamika nakładów inwestycyjnych w tych sektorach. Wyznaczono, a następnie zestawiono w tabeli 2 historyczne wartości dynamiki (względnej zmiany w stosunku do roku poprzedniego) produkcji globalnej (*PG*), nakładów inwestycyjnych (*NI*), zużycia wód powierzchniowych (W_{pw}) i podziemnych (W_{pd}).

Do modelowania zmienności współczynników zużycia wód wybrano modele wykorzystujące wnioskowanie rozmyte w architekturze Mamdaniego. Każdą ze zmiennych opisano przez 5 wartości lingwistycznych, którym przypisano odpowiednie zbiory rozmyte o funkcjach przynależności typu gaussowskiego. Zastosowano mechanizm inferencji MAX–MIN i metodę defuzyfikacji – metodę środka ciężkości. Przyjęto następujący schemat modelu:

Modele strojono metodą prób i błędów, dążąc do minimalizacji sumy kwadratów odchyleń pomiędzy wartościami historycznymi współczynników wodochłonności i wartościami otrzymanymi z modeli. Poniżej przedstawiono oszacowane modele zmienności współczynnika wodochłonności wód powierzchniowych dla sektorów A i B (przykładowe 2 z 12 estymowanych modeli).

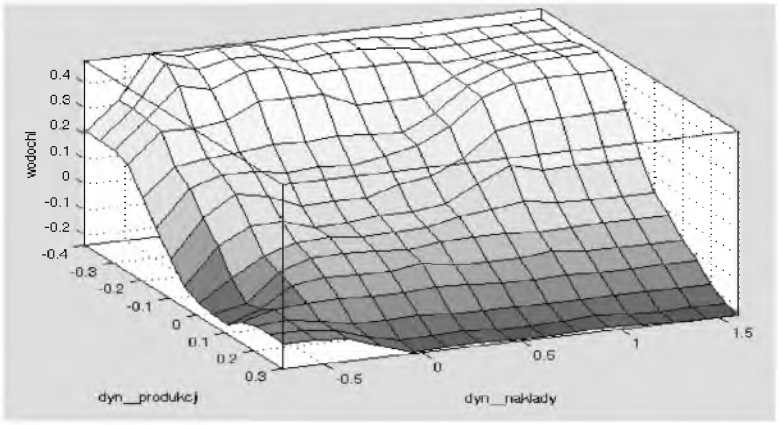


Rysunek 1. Model Mamdaniego dynamiki wodochlonności wód powierzchniowych w sektorze A

Tabela 2. Zmienność (w stosunku do roku poprzedniego) *PG*, *NI* oraz W_{pw} i W_{pd} w poszczególnych sektorach gospodarki w latach 1993–2004

Sektor	Zmienna	1993	1994	1995	1996	1997	1998	1999	2000	2001	2002	2003	2004
A	*PG*	0,015	–0,041	0,157	0,012	–0,105	–0,032	–0,093	–0,023	0,021	–0,066	–0,014	0,056
	NI	–0,017	0,058	0,237	0,311	–0,051	–0,188	0,009	–0,037	–0,092	–0,016	–0,025	0,161
	W_{pw}	–0,108	–0,181	–0,178	–0,357	–0,072	0,062	–0,102	0,231	–0,261	0,126	–0,001	–0,054
B	*PG*	–0,297	–0,014	–0,171	0,041	0,011	–0,104	0,047	–0,176	–0,253	–0,344	0,283	–0,203
	NI	0,802	–0,544	–0,688	1,632	–0,218	0,242	0,305	–0,087	–0,280	–0,233	–0,042	0,562
	W_{pw}	0,460	–0,050	0,147	–0,087	0,040	0,017	0,027	0,210	0,339	0,451	–0,120	0,331
C	*PG*	0,037	0,143	–0,029	–0,019	0,042	–0,150	–0,043	–0,006	–0,041	–0,014	–0,018	0,183
	NI	–0,062	0,121	–0,053	–0,060	–0,017	0,022	–0,004	–0,215	0,183	–0,046	–0,014	0,059
	W_{pw}	–0,053	–0,123	0,066	–0,130	–0,127	–0,489	–0,032	–0,031	–0,152	–0,132	0,094	–0,420
	W_{pd}	–0,306	–0,129	0,228	0,031	–0,171	–0,347	0,053	–0,009	–0,049	0,388	–0,387	–0,254
D	*PG*	0,031	0,105	0,055	0,049	0,093	0,025	0,011	0,063	–0,034	–0,005	0,104	0,152
	NI	–0,125	0,355	0,142	0,254	0,179	0,152	–0,077	–0,113	–0,134	–0,015	0,117	0,134
	W_{pw}	–0,137	–0,118	–0,050	–0,104	–0,129	–0,112	–0,142	–0,056	–0,152	0,002	–0,416	0,412
	W_{pd}	–0,110	–0,136	–0,119	–0,070	–0,079	–0,154	–0,126	–0,171	–0,092	–0,039	–0,112	–0,138
E-en.	*PG*	–0,111	0,007	0,000	–0,012	0,023	–0,007	0,115	0,045	0,182	0,054	0,044	–0,025
	NI	0,141	0,097	0,223	0,135	–0,001	–0,039	–0,009	–0,203	0,149	–0,005	–0,127	–0,061
	W_{pw}	0,116	–0,004	0,047	0,046	–0,032	–0,003	–0,128	–0,066	–0,159	–0,026	–0,015	0,029
	W_{pd}	0,065	–0,084	–0,021	–0,071	0,013	–0,054	0,046	–0,016	–0,193	–0,090	0,006	0,038
E-pob.	*PG*	0,144	0,046	–0,189	0,028	0,068	0,037	0,105	0,222	0,041	0,136	0,012	–0,035
	NI	0,170	0,083	–0,066	0,026	0,027	–0,145	–0,077	–0,081	–0,051	–0,126	–0,024	0,220
	W_{pw}	–0,161	–0,128	0,111	–0,077	–0,108	–0,094	–0,127	–0,222	–0,120	–0,178	–0,021	–0,019
	W_{pd}	–0,149	–0,060	0,213	–0,003	–0,123	–0,069	0,078	–0,178	–0,078	–0,117	0,005	0,006
F-O	*PG*	0,025	0,009	0,127	0,122	0,121	0,111	0,090	0,096	0,032	0,023	0,011	0,025
	NI	0,004	0,012	0,148	0,189	0,309	0,197	0,125	0,076	–0,116	–0,137	–0,026	0,043
	W_{pw}	0,853	0,356	1,731	–0,443	–0,340	–0,513	0,422	0,001	–0,145	0,043	0,054	–0,010
	W_{pd}	–0,035	0,007	–0,150	–0,184	–0,094	–0,175	–0,073	–0,202	–0,123	–0,111	0,012	0,001

W przypadku sektora A model opisuje zmienność wodochłonności w następujący sposób: wodochłonność spada o 25–35% przy wzroście produkcji rzędu 10–20% lub wzroście nakładów o 15–35%; wodochłonność wzrasta o 10–15% przy spadku produkcji o 5–15% i jednoczesnym spadku nakładów inwestycyjnych o 5–20%.



Rysunek 2. Model Mamdaniego dynamiki wodochłonności wód powierzchniowych w sektorze B

W sektorze B wysoki wzrost współczynnika wodochłonności powoduje głównie spadek produkcji, niemal niezależnie od poziomu zmian w nakładach inwestycyjnych.

W podobny sposób można scharakteryzować zależności opisujące modele zmian współczynników wodochłonności wód powierzchniowych i podziemnych pozostałych sektorów. W modelach tych najczęściej zakładano:

- spadek wodochłonności przy wysokim wzroście dynamiki produkcji, co można interpretować jako pełniejsze wykorzystanie zdolności produkcyjnych – sektor C (wody podziemne), sektor D (wody powierzchniowe i podziemne), sektor E-en. (wody powierzchniowe i podziemne), sektor E-pob. (wody powierzchniowe i podziemne), sektory F–O (wody podziemne);
- spadek wodochłonności przy wysokim wzroście nakładów, które w pewnym zakresie są także przeznaczane na racjonalizację technologii produkcji – sektor C (wody podziemne), sektor D (wody powierzchniowe i podziemne), sektor E-en. (wody powierzchniowe i podziemne), sektor E-pob. (wody powierzchniowe i podziemne), sektory F–O (wody powierzchniowe i podziemne);
- wzrost wodochłonności przy spadku dynamiki produkcji, zauważalny we wszystkich modelach z wyjątkiem modelu współczynnika wód powierzch-

niowych sektora C, który przy zdecydowanych spadkach w produkcji generuje równoczesne spadki w wodochłonności wód powierzchniowych.

Aby ocenić jakość modeli, obliczono na podstawie danych historycznych i wartości otrzymanych z modeli:

- średni błąd kwadratowy według wzoru:

$$\delta = \frac{\left(b - \tilde{b}\right)^2}{n},$$

gdzie:

b – wartości rzeczywiste współczynnika wodochłonności,
$\tilde{b}$ – wartości współczynnika wodochłonności otrzymane z modelu,
n – liczba obserwacji,

- średni błąd względny według wzoru:

$$s = \frac{\sqrt{\delta}}{\bar{b}},$$

gdzie:

$\bar{b}$ – średni współczynnik wodochłonności (z danych historycznych).

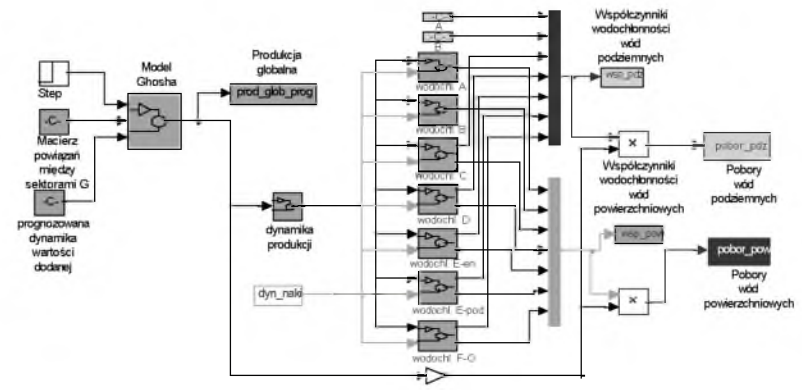
Tabela 3. Średni błąd kwadratowy i błąd względny dla poszczególnych technik modelowania wodochłonności w sektorach gospodarki

Lp.	Sektor		Modele współczynników zużycia wód powierzchniowych		Modele współczynników zużycia wód podziemnych	
			średni błąd kwadratowy	średni błąd względny	średni błąd kwadratowy	średni błąd względny
1.	Rolnictwo	A	0,025	10%		
2.	Rybactwo	B	26186	12%		
3.	Górnictwo	C	0,032	15%	0,0108	21%
4.	Przetwórstwo przemysłowe	D	0,028	12%	0,0010	6%
5.	Energetyka	E-en.	25	4%	0,00088	6%
6.	Pobór wód	E-pob.	523	12%	340	7%
7.	Pozostałe	F–O	0,000004	41%	0,0000234	7%

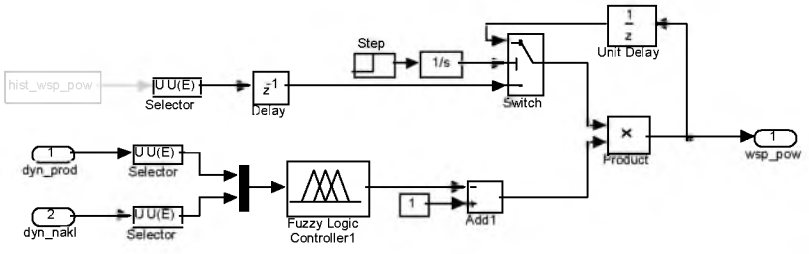
Modele oparte na wnioskowaniu rozmytym dają, biorąc pod uwagę wartości obliczonych błędów, akceptowalne wyniki. Wśród oszacowanych 12 modeli tylko w dwóch przypadkach – modeli współczynników wód powierzchniowych dla grupy sektorów F–O i współczynników wód podziemnych sektora C – błędy były wysokie: 41% i 21%. W pozostałych przypadkach dopasowanie modeli rozmytych do danych historycznych jest dobre – na poziomie od 4 do 15%.

5. Modelowanie i prognoza zużycia wód powierzchniowych i podziemnych przy wykorzystaniu modelu *input-output* i sektorowych modeli dynamiki wodochłonności Mamdaniego

Modele dynamiki współczynników zużycia wód powierzchniowych i podziemnych przez poszczególne sektory zostały następnie połączone z modelem *input-output* i użyte do prognozy zużycia zasobów wodnych w latach 2005–2015. Schemat modelu przedstawiono na rysunku 3.



Rysunek 3. Schemat modelu *input-output* powiązanego z sektorowymi modelami dynamiki wodochłonności



Rysunek 4. Schemat modułu obliczającego współczynnik wodochłonności na podstawie modelu rozmytego dynamiki wodochłonności

Zmiennymi wejściowymi do tak stworzonego modelu prognozującego pobory wody są prognoza dynamiki wartości dodanej (wymagana do modelu Ghosha) i prognoza dynamiki nakładów inwestycyjnych (do modeli Mamdaniego). Druga ze zmiennych wejściowych do modeli Mamdaniego – dynamika produkcji globalnej – jest wyznaczana na podstawie prognozy produkcji globalnej obliczanej w modelu Ghosha. Dane na temat prognozy dynamiki tych wielkości zaczerpnięto z rządowych planistycznych i strategicznych dokumentów i programów.

Tabela 4. Prognoza dynamiki zmiennych wejściowych w latach 2005–2015

Lp.	Sektor		Dynamika wartości dodanej [%]	Dynamika nakładów inwestycyjnych [%]
1.	Rolnictwo	A	2	5
2.	Rybactwo	B	2	5
3.	Górnictwo	C	–2	0
4.	Przetwórstwo przemysłowe	D	5	5
5.	Energetyka	E-en.	3	1
6.	Pobór wód	E-pob.	1	2
7.	Pozostałe	F–O	5	1

W wyniku modelowania przy tak przyjętych wartościach dynamiki nakładów inwestycyjnych otrzymano następujące prognozy współczynników wodochłonności (tabela 5).

Tabela 5. Prognozowane współczynniki wodochłonności wód powierzchniowych i podziemnych

Lp.	Sektor		Dane historyczne	Prognozowane współczynniki zużycia wód powierzchniowych [m^3/tys. zł]			2015/ 2004
			2004	2007	2010	2015	
1.	Rolnictwo	A	1,1919	0,9864	0,8113	0,5859	49%
2.	Rybactwo	B	2 494,1772	2 982,4407	2 792,3537	2 497,3492	100%
3.	Górnictwo	C	0,3530	0,0906	0,0804	0,0633	18%
4.	Przetwórstwo przemysłowe	D	0,8118	0,7867	0,5108	0,2483	31%
5.	Energetyka	E-en.	110,4739	117,4366	106,5322	89,3927	81%
6.	Pobór wód	E-pob.	100,0683	86,2777	71,0877	51,4637	51%
7.	Pozostałe	F–O	0,0033	0,0034	0,0036	0,0039	116%

Lp.	Sektor		Dane historyczne	Prognozowane współczynniki zużycia wód podziemnych [m³/tys. zł]			2015/ 2004
			2004	2007	2010	2015	
1.	Rolnictwo	A	0,0000	0,0000	0,0000	0,0000	–
2.	Rybactwo	B	0,0000	0,0000	0,0000	0,0000	–
3.	Górnictwo	C	0,2483	0,1565	0,1447	0,1237	50%
4.	Przetwórstwo przemysłowe	D	0,2438	0,1611	0,1125	0,0619	25%
5.	Energetyka	E-en.	0,3994	0,4136	0,3597	0,2814	70%
6.	Pobór wód	E-pob.	200,7053	193,2278	181,0069	162,2647	81%
7.	Pozostałe	F–O	0,0385	0,0301	0,0208	0,0113	29%

W poszczególnych sektorach otrzymano następujące wyniki modelowania zmienności współczynników wodochłonności wód powierzchniowych i podziemnych:

1. Sektor A:
- zmiany historyczne współczynnika wodochłonności wód powierzchniowych: w latach 1992–2004 spadek o 67% (w latach 2000–2004 o 21%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 51%.

2. Sektor B:
- zmiany historyczne współczynnika wodochłonności wód powierzchniowych: w latach 1992–2004 wzrost o 295% (w latach 2000–2004 o 107%), model – w 2015 roku bez zmian w stosunku do 2004 roku (0%), wnioski: w tym sektorze istnieje wyraźny ciągły wzrost wodochłonności związany z jego rozwojem i zmianami w charakterze hodowli – rozwija się akwakultura pstrąga o wysokich wartościach wodochłonności; model wodochłonności zbudowany na podstawie tych danych powinien dawać prognozy dalszych wzrostów.

3. Sektor C:
- zmiany współczynnika wodochłonności wód powierzchniowych: w latach 1992–2004 spadek o 85% (w latach 2000–2004 o 53%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 82%,
- zmiany współczynnika wodochłonności wód podziemnych: w latach 1992–2004 spadek o 74% (w latach 2000–2004 o 40%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 52%, wnioski: w tym sektorze istnieje wyraźny spadek wodochłonności związany z regresją gospodarczą górnictwa, wyniki otrzymane w modelach, mówiące o spadkach o 80 i 50% w horyzoncie prognozy 11 lat, są prawdopodobnie obarczone błędem (modele te nie są dobrze dopasowane), ale

na pewno w najbliższych latach można się spodziewać dalszych spadków wodochłonności w tym sektorze.

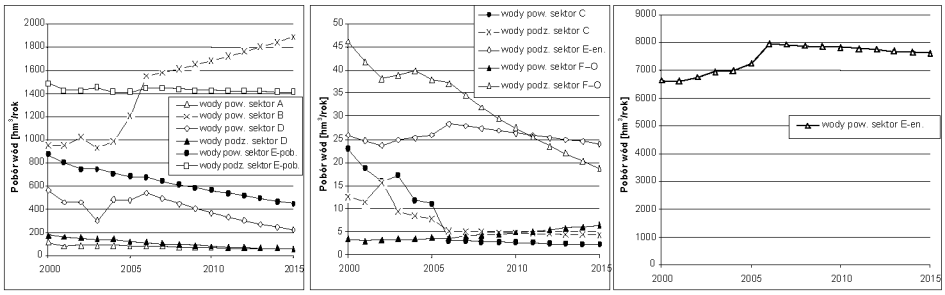
4. Sektor D:
- zmiany współczynnika wodochłonności wód powierzchniowych: w latach 1992–2004 spadek o 72% (w latach 2000–2004 o 30%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 69%,
- zmiany współczynnika wodochłonności wód podziemnych: w latach 1992–2004 spadek o 76% (w latach 2000–2004 o 33%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 75%.

5. Sektor E-en.:
- zmiany współczynnika wodochłonności wód powierzchniowych: w latach 1992–2004 spadek o 21% (w latach 2000–2004 o 17%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 19%,
- zmiany współczynnika wodochłonności wód podziemnych: w latach 1992–2004 spadek o 33% (w latach 2000–2004 o 23%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 27%.

6. Sektor E-pob.:
- zmiany współczynnika wodochłonności wód powierzchniowych: w latach 1992–2004 spadek o 71% (w latach 2000–2004 o 30%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 49%,
- zmiany współczynnika wodochłonności wód podziemnych: w latach 1992–2004 spadek o 42% (w latach 2000–2004 o 18%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 10%, wnioski: wartość sumarycznego poboru (z wód powierzchniowych i podziemnych) wyniosła 1860 hm^3/rok, a po przeliczeniu na jednego mieszkańca (przy prognozie ludności 37 626 tys. osób) – 136 l/d; ten sam wskaźnik obliczony dla 2004 roku wyniósł 152 l/d, dlatego też otrzymaną projekcję poboru tego sektora można uznać za akceptowalną.

7. Sektory pozostałe F–O:
- zmiany współczynnika wodochłonności wód powierzchniowych: w latach 1992–2004 wzrost o 63% (w latach 2000–2004 spadek o 7%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 16%,
- zmiany współczynnika wodochłonności wód podziemnych: w latach 1992–2004 spadek o 71% (w latach 2000–2004 o 21%), model – w 2015 roku spadek (w stosunku do 2004 roku) o 67%.

Przedstawione powyżej prognozy współczynników wodochłonności zostały następnie użyte do wyznaczenia prognoz poborów wód powierzchniowych i podziemnych w poszczególnych sektorach gospodarki (rysunek 5 i tabela 6).



Rysunek 5. Prognoza poborów wód powierzchniowych i podziemnych w sektorach gospodarki

Tabela 6. Prognozowany pobór wód powierzchniowych i podziemnych

Lp.	Sektor		Dane historyczne	Prognozowany pobór wód powierzchniowych [hm³/rok]			2015/ 2004
			2004	2007	2010	2015	
1.	Rolnictwo	A	86,20	76,07	69,04	59,03	68%
2.	Rybactwo	B	985,20	1578,17	1686,45	1885,49	191%
3.	Górnictwo	C	11,80	2,94	2,63	2,14	18%
4.	Przetwórstwo przemysłowe	D	485,50	492,18	366,93	225,03	46%
5.	Energetyka	E-en.	6971,10	7905,30	7805,51	7607,97	109%
6.	Pobór wód	E-pob.	703,70	643,99	560,90	447,95	64%
7.	Pozostałe	F–O	3,46	3,89	4,71	6,47	187%
	OGÓŁEM		9246,96	10 702,53	10 496,16	10 234,09	111%
Lp.	**Sektor**		**Dane historyczne**	**Prognozowany pobór wód podziemnych [hm³/rok]**			**2015/ 2004**
			2004	**2007**	**2010**	**2015**	
1.	Rolnictwo	A	0,00	0,00	0,00	0,00	–
2.	Rybactwo	B	0,00	0,00	0,00	0,00	–
3.	Górnictwo	C	8,30	5,07	4,73	4,18	50%
4.	Przetwórstwo przemysłowe	D	145,80	100,75	80,80	56,15	39%
5.	Energetyka	E-en.	25,20	27,84	26,36	23,95	95%
6.	Pobór wód	E-pob.	1 411,40	1 442,28	1 428,20	1 412,39	100%
7.	Pozostałe	F–O	39,91	34,37	27,41	18,82	47%
	OGÓŁEM		1 630,61	1 610,32	1 567,50	1 515,49	93%

Modele dynamiki współczynników wodochłonności prognozują na 2015 rok spadki współczynników dla 10 z 12 sektorów. Natomiast prognozy wartości poborów mają bardziej zróżnicowane tendencje, powodowane z jednej strony spadkiem współczynników wodochłonności, a z drugiej wzrostem produkcji. Pobór wód powierzchniowych będzie w 2015 roku o około 11% wyższy niż obecnie – najwyższy wzrost (o 90%) nastąpi w sektorze B, w sektorach F–O wzrost wyniesie 87%, a w sektorze E-en. – 9%. W sektorze B będzie on powodowany tylko wzrostem produkcji, bo współczynnik wodochłonności w 2015 roku ma być na poziomie wartości z 2004 roku, natomiast w sektorze E-en. wzrost o 9% wystąpi przy obniżeniu wodochłonności o 20% (jeśli wodochłonność nie ulegnie takiemu obniżeniu, to można się spodziewać wyższych poborów). Wspomniany wzrost poborów wód powierzchniowych w sektorach F–O jest przewidywany przy wzroście współczynnika wodochłonności o 16%. W pozostałych sektorach nastąpi obniżenie poborów wód powierzchniowych – o około 32% w sektorze A, o około 80% – w sektorze C, o 54% w sektorze D i o 34% w sektorze E-pob.

W przypadku wód podziemnych spadki w wodochłonności spowodują, pomimo wzrostu produkcji, obniżenie poborów – w całej gospodarce o około 7%. Najbardziej, bo o 60%, obniży pobór wód podziemnych sektor D, o 50% – sektor C i sektory F–O, a o 5% sektor E-en. Natomiast pobory sektora E-pob. będą na poziomie wartości z 2004 roku.

6. Wnioski

1. Analiza danych historycznych dotyczących poborów wody, produkcji oraz współczynników zużycia wód dla potrzeb produkcji wykazuje dużą zmienność wodochłonności wszystkich sektorów gospodarki, której nie można pomijać przy prognozowaniu zapotrzebowania na wodę. (Dane statyczne, w tym najnowsze, pokazują, że w wielu sektorach wzrostowi produkcji towarzyszy spadek poborów wody).
2. Podjęta próba modelowania wodochłonności poszczególnych sektorów gospodarki przy użyciu technik wykorzystujących wnioskowanie rozmyte wypadła pozytywnie. Przyjęta technika wnioskowania potwierdza istnienie zmienności współczynników wodochłonności w czasie i daje możliwość jej zamodelowania. Zmiany wodochłonności mają dla większości sektorów gospodarki charakter spadkowy. Zastosowane techniki pozwoliły na sformułowanie modeli tych zmian. Jakość tych modeli – dopasowanie do danych historycznych oraz jakość otrzymywanych dzięki nim prognoz – jest w dużej mierze pochodną niskiej liczebności zbioru danych historycznych, na których modele były estymowane.

3. Połączenie modelu *input-output* z modelami zmienności wodochłonności poszczególnych sektorów to połączenie dwóch wielkości – prognozy produkcji i prognozy wodochłonności. Iloczyn tych dwóch wielkości daje prognozę zapotrzebowania na wodę w podziale na sektory, uwzględniającą rozwój gospodarczy i sektorową dynamikę wodochłonności.

Bibliografia

[1] Duarte R., Sánchez-Chóliz J., Bielsa J., *Water Use in the Spanish Economy: An Input-Output Approach*, „Ecological Economics" 2002, No. 43.
[2] Gurgul H., *Modele input-output w warunkach niepełnej informacji*, Kraków 1998.
[3] Kasprzyk J., *Wieloetapowe sterowanie rozmyte*, Warszawa 2001.
[4] Lenzen M., Foran B., *An Input-Output Analysis of Australian Water Usage*, „Water Policy" 2001, No. 3.
[5] Łachwa A., *Rozmyty świat zbiorów, relacji i reguł*, Warszawa 2001.
[6] Piegat A., *Modelowanie i sterowanie rozmyte*, Warszawa 1999.
[7] Plich M., *Budowa i zastosowanie wielosektorowych modeli ekonomiczno-ekologicznych*, Łódź 2002.
[8] Velazquez E., *An Input-Output Model of Water Consumption: Analysing Intersectoral Water Relationships in Andalusia*, „Ecological Economics" 2006, No. 56.

Renata Uryga
Barbara Mrzygłód
Agnieszka Smolarek-Grzyb

Ontologiczna reprezentacja wiedzy

1. Wstęp

Potrzeba integracji wiedzy rozproszonej oraz udostępniania jej zarówno maszynom, jak i człowiekowi w postaci zrozumiałej, a jednocześnie nadającej się do automatycznego przetwarzania przyczyniła się do intensyfikacji rozwoju wielu dziedzin nauki: matematyki, logiki, sztucznej inteligencji, inżynierii wiedzy itd. Upowszechnienie zastosowań komputerów w różnych dziedzinach działalności człowieka wymusza poszukiwanie i konstruowanie narzędzi matematycznych, logicznych oraz informatycznych, które pozwolą na tworzenie reprezentacji wiedzy z różnych dziedzin, nadających się do wykorzystania w komputerowych systemach wspomagania decyzji, systemach ekspertowych, systemach sterujących urządzeniami bądź procesami, np. technologicznymi. Takimi formalizmami mogą być np. dowolne logiki, sieci Bayesa, sieci semantyczne. Wybór formalizmu zależy od typu rozwiązywanego problemu.

Globalnym podejściem do problematyki reprezentacji wiedzy w systemach komputerowych odznaczają się ontologie [1]. Początkowo służyły one do modelowania danych, ale wraz z postępem techniki, rozwojem informatyki, sieci komputerowych, sztucznej inteligencji itd. dostrzeżono możliwość wykorzystania ich do usprawnienia dystrybucji wiedzy.

2. Pojęcia podstawowe

Przez ontologię rozumie się [2] formalną specyfikację wspólnej warstwy pojęciowej, gdzie warstwą pojęciową nazywany jest pewien abstrakcyjny model zjawisk występujących w pewnym ograniczonym wycinku rzeczywistości, który otrzymuje się przez identyfikację występujących w nim istotnych pojęć (nazwy obiektów, zdarzeń, stanów) oraz wyszczególnienie relacji pomiędzy nimi. Model musi być czytelny dla maszyn: „formalizacja wiedzy", pojęcia muszą być zdefiniowane jednoznacznie, „specyfikacja", na koniec wiedza zawarta w ontologii

powinna być „wspólna", czyli akceptowana przez wszystkich użytkowników (maszyny i ludzi). W takim rozumieniu ontologia jest po prostu pewnym modelem wycinka rzeczywistości, zapisanym w postaci zrozumiałej dla maszyn przetwarzających dane i informacje. Podstawowymi komponentami tego modelu są pojęcia (ang. *concepts*) odpowiadające obiektom. Pojęcia zgrupowane są w klasy oraz podklasy i tworzą pewną uporządkowaną hierarchię. Każdy obiekt posiada cechy swojej klasy i klas nadrzędnych.

Najczęściej jako podstawę budowy modelu wykorzystuje się ramy typu: klasy, sloty, fasety, instancje:

- **klasy** to zbiory podstawowych pojęć z dziedziny;
- **sloty** przechowują pewne własności i cechy obiektów z klasy, posiadają one ograniczenia w postaci *fasetów*, np. slot „zawartość węgla" dla klasy „stal" może posiadać faset – ograniczenie „max. 2,11%";
- **instancje** – zdefiniowane obiekty danej klasy, np. klasa „stale" posiada instancję „stal St3s", dla której za pomocą slotów został zdefiniowany konkretny skład chemiczny i stosowane procesy obróbki.

Tworzenie ontologii jest złożone i wieloetapowe. Bassara [3] motywację postrzega jako proces inicjujący. Motywacją do tworzenia ontologii może być:

- chęć dzielenia się wspólnym rozumieniem ustrukturalizowanej informacji zarówno między ludźmi, jak i maszynami (agentami internetowymi);
- umożliwienie wielokrotnego wykorzystania wiedzy z danej dziedziny;
- uczynienie założeń danej dziedziny wiedzy bardziej oczywistymi;
- oddzielenie dziedziny wiedzy od działań operacyjnych w danej dyscyplinie;
- analizowanie danej dziedziny wiedzy.

W dalszej części artykułu przedstawiono ontologiczną reprezentację wiedzy o wyrobach metalowych zaimplementowaną w systemie komputerowym. Autorki zostały zmotywowane do jej stworzenia nadzieją, że przy ontologicznym ujmowaniu trudnej wiedzy z zakresu metaloznawstwa rozumienie i uczynienie bardziej przejrzystymi jej założeń oraz występujących relacji zaowocuje podniesieniem efektywności nauczania w tej dziedzinie.

3. Ontologiczna reprezentacja wiedzy o wyrobach metalowych

3.1. Wiedza o metalach i ich stopach

Wiedza o metalach i ich stopach jest zawarta w bogatej literaturze metaloznawczej, w normach, procesach technologicznych [4, 5, 6]. Cechą charaktery-

styczną tej wiedzy jest niejednorodność formy, stosowanie różnych kryteriów jej porządkowania. Wydobycie wiedzy potrzebne do rozwiązania konkretnego problemu (np. wytypowania przyczyn powtarzania wad wyrobów w konkretnym procesie) wymaga ogromnego nakładu pracy, zwykle dużej grupy ludzi. Uporządkowanie tej wiedzy według jasno określonego klucza znacznie ułatwia dostęp do danych jej obszarów. Na rysunku 1 pokazano hierarchiczny sposób porządkowania wiedzy o metalach i ich stopach. Taki porządek ułatwia zarówno wydobywanie wiedzy, jak i edukację w danym zakresie tematycznym.

Rysunek 1. Ogólny podział metali i ich stopów

Analizując wiedzę z wybranej dziedziny, stwierdzono hierarchiczny układ występujących w niej pojęć kluczowych, w którym pojęcia niżej usytuowane dziedziczyły cechy pojęć nadrzędnych. Hierarchizacja pojęć w danej dziedzinie wprowadza w niej pewien porządek, bardzo wyraziście przedstawia relacje uogólniania i uszczegóławiania w wybranym zbiorze pojęć.

3.2. Edytor ontologii

Do stworzenia ontologicznej reprezentacji fragmentu wiedzy metaloznawczej wykorzystano pakiet Protégé-2000 [7].

Edytor Protégé-2000 powstał 15 lat temu w Departamencie Informatyki Medycznej Uniwersytetu Stanford. W pierwotnej wersji Protégé-2000 był specjalistycznym programem wspomagającym aktywizację wiedzy dla systemów ekspertowych w dziedzinie medycyny. Obecnie jest używany do tworzenia ontologii

z różnych dziedzin oraz – na ich podstawie – baz wiedzy. Edytor jest narzędziem darmowym, a językiem implementacji jest Java. Model wiedzy edytora opiera się na ramach. W ramowych modelach reprezentacji wiedzy pojęcia reprezentowane są przez zbiory obiektów o wspólnych właściwościach.

3.3. Prezentacja ontologii

Skonstruowana ontologia metali i ich stopów składa się z dwóch rozdzielnych modułów:

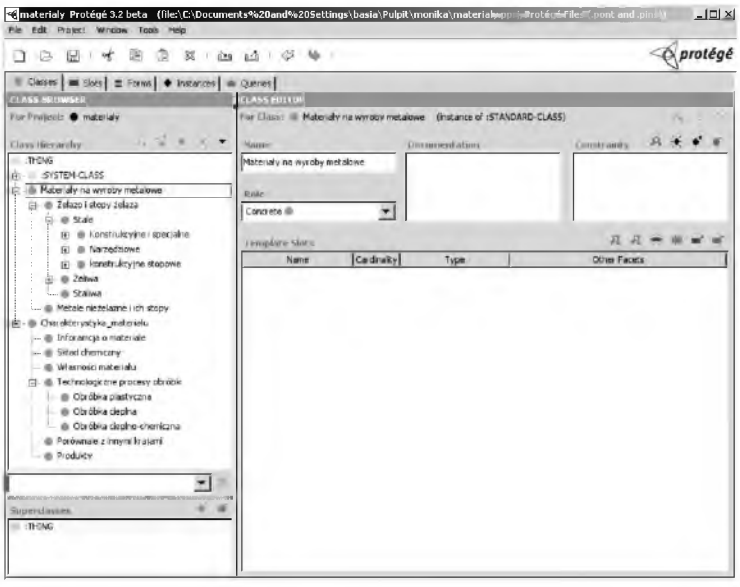
- **materiały na wyroby metalowe** (zastosowano klasyfikację ze względu na zastosowanie według Dobrzańskiego [8, 9]);
- **charakterystyka materiału.**

Modularna budowa modelu jest jednym z podstawowych warunków umożliwiających wielokrotne użycie wiedzy [10].

W dalszej części pracy zaprezentowano zrzuty ekranowe ilustrujące rezultaty pracy polegającej na tworzeniu ontologicznej reprezentacji dwóch podstawowych klas.

3.3.1. Klasa „materiały na wyroby metalowe"

Klasa „materiały na wyroby metalowe" charakteryzuje się strukturą (hierarchią) taką samą jak zaproponowano na rysunku 1.

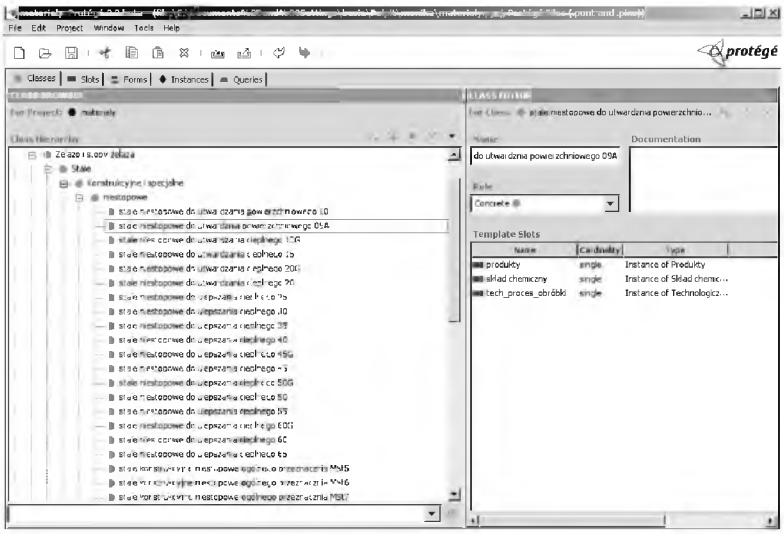


Rysunek 2. Klasa „materiały na wyroby metalowe"

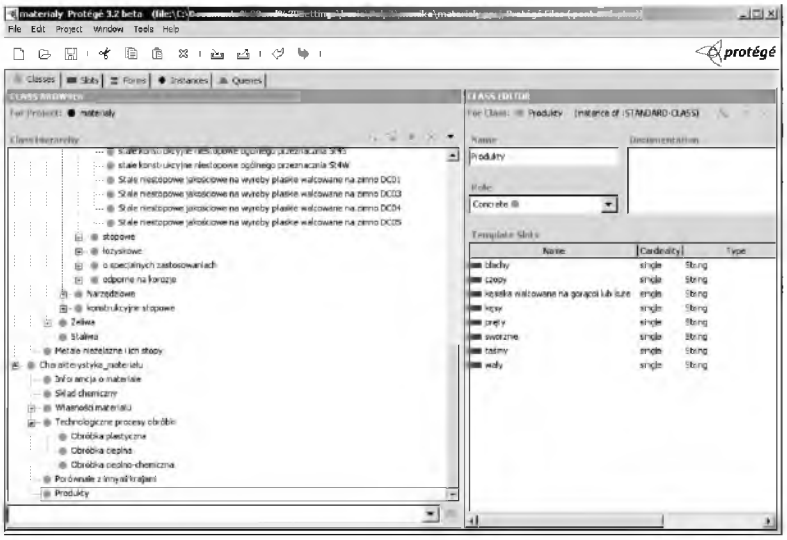
W ramach tej klasy definiowane są podklasy przedstawione na rysunku 3. Ten sposób klasyfikacji jest zgodny z przyjętym przez Dobrzańskiego [4, 5, 6].

Rysunek 3. Przykład hierarchii podklas w klasie „materiały na wyroby metalowe”

Przykładem podklasy są np. „stale niestopowe do utwardzania powierzchniowego" pokazane na rysunku 4. Niniejszej podklasie przypisano 3 sloty, które ją charakteryzują („produkty", „skład chemiczny", „tech_proces_obrobki"). Są to sloty typu instancja, zawierające odwolania do konkretnych klas (typ obiektowy).



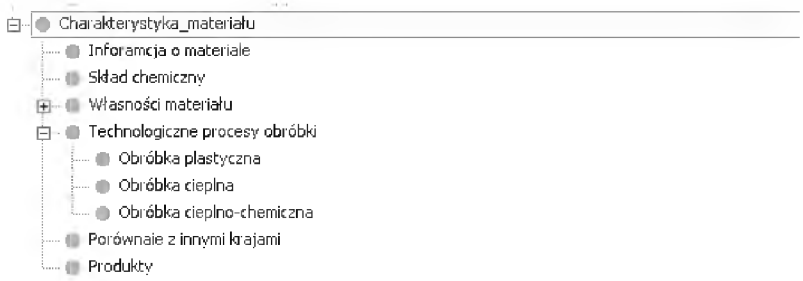
Rysunek 4. Przykład podklasy „stale niestopowe do utwardzania powierzchniowego" ze slotami typu instancje



Rysunek 5. Przykładowe instancje stworzone dla produktów stali niestopowych

3.3.2. Klasa „charakterystyka materiału”

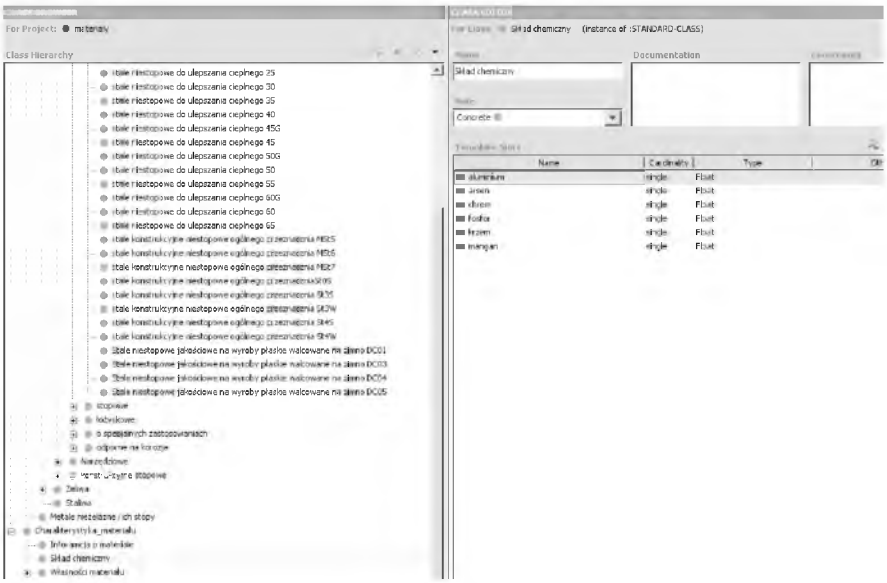
Na kolejnych rysunkach pokazano konstrukcję klasy „charakterystyka materiału”. Posiada ona strukturę (hierarchię) zaprezentowaną na rysunku 6.



Rysunek 6. Klasa „charakterystyka materiału”

Klasa „charakterystyka materiału” składa się z 4 podklas:
- informacje o materiale,
- skład chemiczny,
- własności materiału,
- technologiczne procesy obróbki.

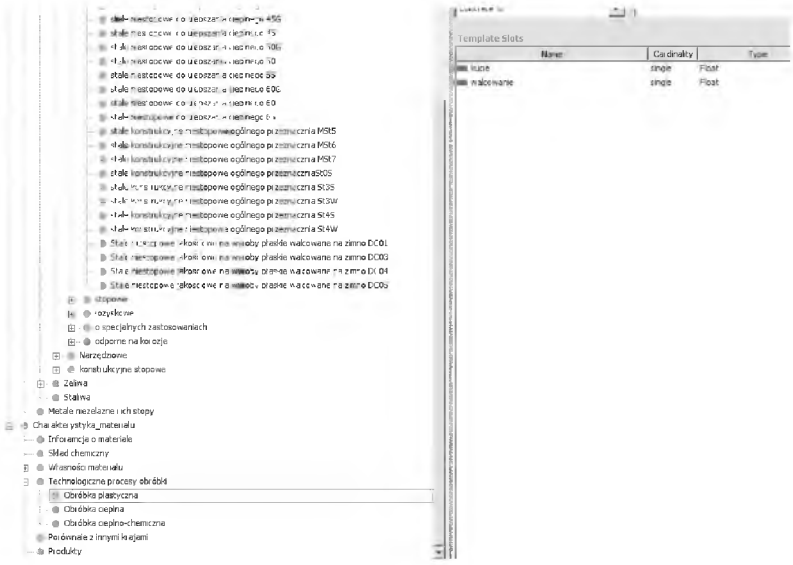
Na rysunku 7 pokazano sloty w podklasie „skład chemiczny”.



Rysunek 7. Sloty w podklasie „skład chemiczny”

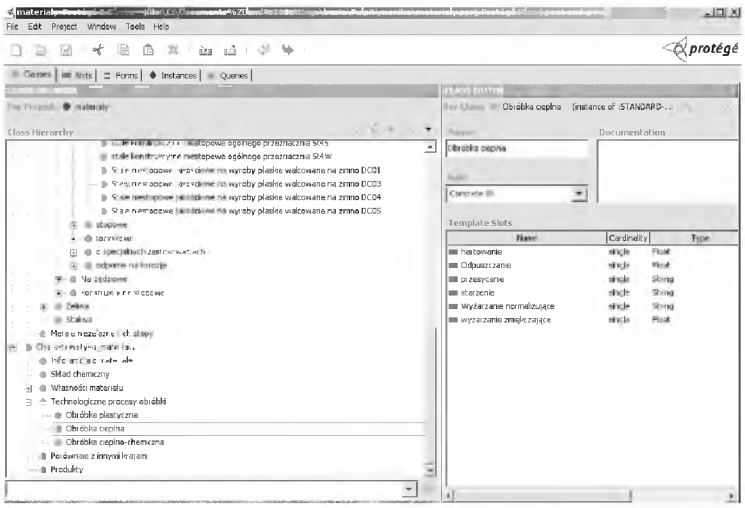
W podklasie „technologiczne procesy obróbek" wyróżniono podklasy:

- „obróbka plastyczna" (sloty: kucie, walcowanie), podklasa ta została przedstawiona na rysunku 8.



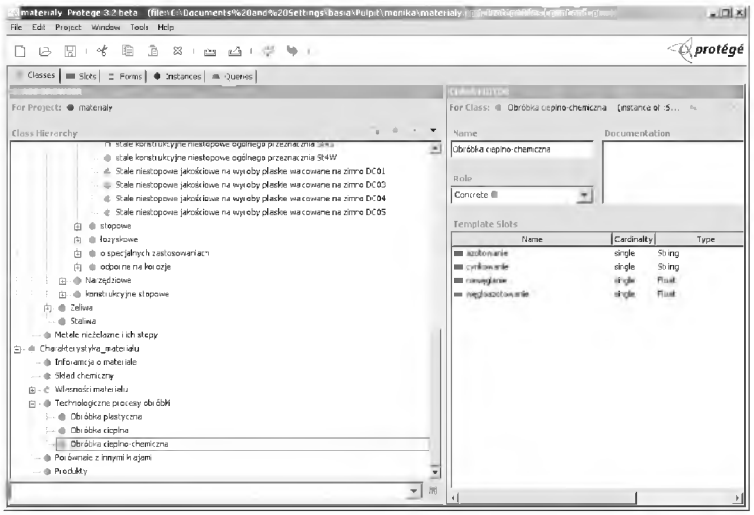
Rysunek 8. Sloty w podklasie „obróbka plastyczna" klasy technologiczne procesy obróbki

- „obróbka cieplna" (sloty: hartowanie, odpuszczanie, przesycanie, starzenie, wyżarzanie normalizujące, wyżarzanie zmiękczające), podklasa ta została przedstawiona na rysunku 9.



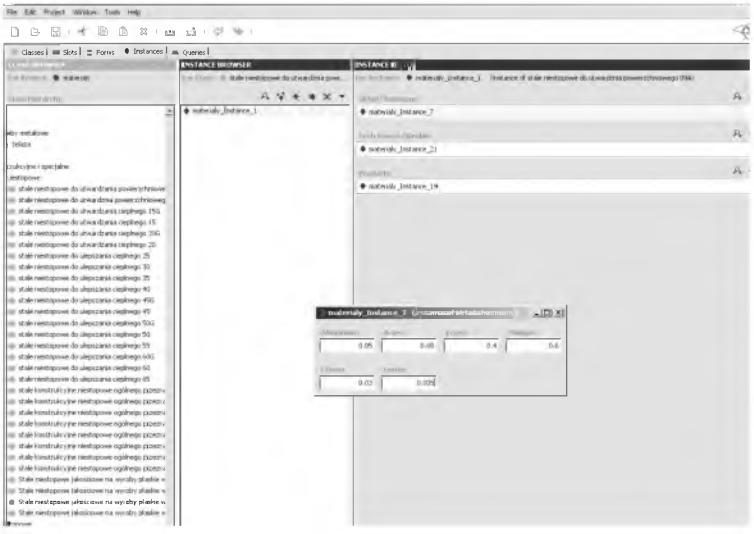
Rysunek 9. Sloty w podklasie „obróbka cieplna" zawierającej się w podklasie „technologiczne procesy obróbki" (klasa „charakterystyka materiału")

• „obróbka cieplno-chemiczna" (sloty: azotowanie, cynkowanie nawęglanie węgloazotowanie), podklasa ta została przedstawiona na rysunku 10.



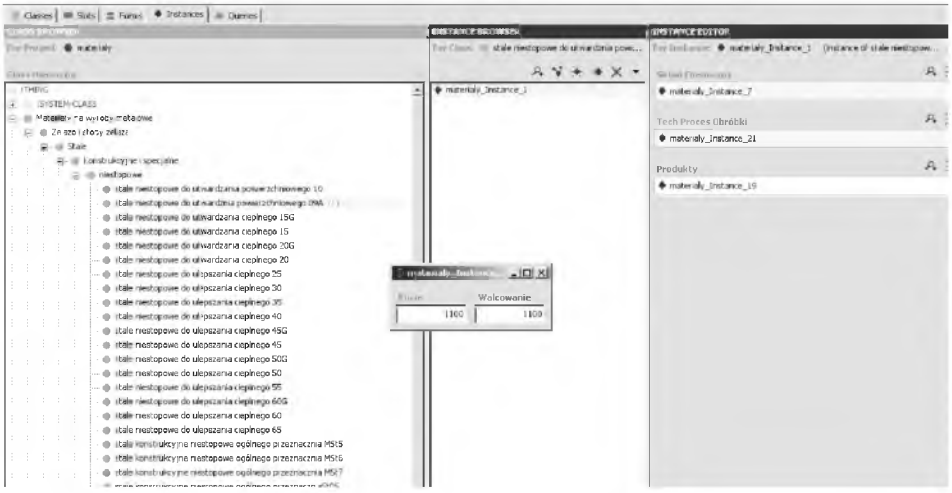
Rysunek 10. Sloty w podklasie „obróbka cieplno-chemiczna" zawierającej się w podklasie „technologiczne procesy obróbki" (klasa „charakterystyka materiału")

Maksymalne wartości slotów dla instancji „materiały – skład chemiczny" dla stali niestopowej do utwardzania powierzchniowego 09A przedstawiono na rysunku 11.



Rysunek 11. Maksymalne wartości slotów dla instancji „materiały – skład chemiczny" dla stali niestopowej do utwardzania powierzchniowego 09A

Maksymalne dopuszczalne temperatury kucia i walcowania dla stali niestopowej do utwardzania powierzchniowego 09A przedstawiono jako sloty instancji „materiały – obróbka plastyczna" na rysunku 12.



Rysunek 12. Maksymalne wartości dopuszczalne temperatury kucia i walcowania dla stali niestopowej do utwardzania powierzchniowego 09A przedstawione jako sloty instancji „materiały – obróbka plastyczna"

Przedstawione fragmenty ontologii ukazują wyraziście zalety tej metody reprezentacji wiedzy. Warto podkreślić klarowność przekazywania wiedzy o relacjach i związkach pomiędzy kluczowymi pojęciami w strukturze hierarchicznej.

4. Podsumowanie i wnioski

Temat niniejszej pracy obejmuje bardzo niewielki wycinek rzeczywistości. Wiedza zawierająca ludzkie poznanie i dokonania w dziedzinie wyrobów metalowych jest obszerna, rozproszona w różnych dziedzinach nauk szczegółowych i przechowywana w różnorakich postaciach. Dostęp do wybranych (potrzebnych do wykonania określonego zadania) komponentów wiedzy jest w związku z tym czasochłonny, utrudniony, bo wymaga dotarcia do rozproszonych (geograficznie) źródeł, i kosztowny.

Reprezentacja wiedzy w formie ontologii może dotyczyć dowolnego fragmentu rzeczywistości (jak to pokazano na przykładzie), jednakże główną zaletą tego przedstawienia jest stworzenie możliwości bardziej uniwersalnego operowania

wiedzą, a w szczególności integracji (łącznego wykorzystania) wiedzy pochodzącej z różnych źródeł.

Obecnie badania dotyczące integracji oraz innych właściwości ontologicznej reprezentacji wiedzy znajdują się wciąż jeszcze w fazie rozwoju. Brakuje też efektywnych narzędzi realizacji procedur wnioskowania (istniejące rozwiązania charakteryzują się dużą złożonością obliczeniową). Rozważania przedstawione w tym opracowaniu są wskazaniem nowych trendów rozwojowych inżynierii wiedzy.

Bibliografia

[1] Gliński W., *Wstęp do ontologii*, Politechnika Warszawska, wykład z 27 października 2005 roku, http://www.icie.com.pl/ZISI/Glinski_Ontologie.pdf.
[2] Gruber T.R., *A Transactional Approach to Portable Ontology Specification*, „Knowledge Acquisition" 1993, 5(2), s. 199–221.
[3] Bassara A., *I weź tu się dogadaj – ontologie*, „Gazeta IT" 2004, nr 1(20).
[4] Dobrzański L., *Metaloznawstwo i obróbka cieplna stopów metali*, Gliwice 1993.
[5] Rudnik S., *Metaloznawstwo*, Warszawa 1986.
[6] Blicharski M., *Wstęp do inżynierii materiałowej*, Kraków 1995.
[7] http://protege.stanford.edu.
[8] Dobrzański L., *Metaloznawstwo i obróbka cieplna stopów metali*, Gliwice 1993.
[9] Dobrzański L., *Leksykon materiałoznawstwa: praktyczne zestawienie norm polskich, zagranicznych i międzynarodowych*, Warszawa 2003.
[10] Rector A.L., *Modularisation of Domain Ontologies Implemented in Description Logics and Related Formalisms Including OWL*, [w:] *Proceedings K-CAP'03, October 23–25*, 2003.

Stanisława Kluska-Nawarecka
Agnieszka Smolarek-Grzyb
Dorota Wilk-Kołodziejczyk

Ontologie w reprezentacji wiedzy o wadach wyrobów odlewniczych

1. Reprezentacja wiedzy na temat wad odlewniczych

Prowadzenie procesu technologicznego w produkcji wyrobów metalowych wymaga wykrywania i definiowania rodzaju wad oraz ustalania przyczyn ich powstawania, aby można im było zapobiegać. Bardzo często stosuje się do tego celu metody sztucznej inteligencji, które wspomagają decyzję i korzystają ze specjalnie utworzonej bazy wiedzy. Inteligentna baza wiedzy oferuje różnorodne metody przedstawiania, wymiany danych i wiedzy technologicznej.

Ważnym zagadnieniem przy tworzeniu bazy wiedzy jest odpowiedni dobór formalizmu do jej reprezentacji. Klasyfikacja metod reprezentacji wiedzy została przedstawiona na rysunku 1 [1, 2].

Metody reprezentacji wiedzy

Grafy i sieci	Formuły logiczne	Macierze i tablice	Modele heurystyczne
• sieci semantyczne, ontologie • grafy wiedzy	• logika rozmyta • logika przybliżona • logiki nieklasyczne: - modalna - temporalna - defaultowa • logika klasyczna: - stwierdzenia, predykaty - dwuwartościowa - wielowartościowa	• wektory i macierze wiedzy • ramy	• algorytmy genetyczne • sztuczne sieci neuronowe • modele automatowe i obliczeniowe

Rysunek 1. Klasyfikacja metod reprezentacji wiedzy
Źródło: na podstawie [1, 2].

W niniejszym artykule przedstawiono charakterystykę etapów budowy ontologii jako bazy wiedzy działającej na rzecz wykrywania przyczyn powstawania wad. System ten ma duże szanse w istotny sposób przyczynić się do podniesienia jakości wyrobów i poprawić efektywność produkcji.

2. Ontologia

Ontologie stanowią formalny i cieszący się dużą popularnością sposób opisu modelu wiedzy. Termin „ontologia" ma swoje korzenie w filozofii, a w kontekście informatycznym pojawił się w 1967 roku w pracach S.H. Mealy'ego dotyczących modelowania danych. Dopiero jednak w dobie morza informacji dostępnych w Internecie i konieczności ich wymiany i przetwarzania zyskał on szersze zainteresowanie. Ontologia zajmuje się reprezentacją i opisywaniem „tego, co jest" – w rzeczywistości, w umysłach ludzi, i zapisanego w postaci różnych symboli (za S.H. Mealym). Ontologia zawsze zajmuje się pewnym fragmentem rzeczywistości – mniej lub bardziej dokładnie określonym. Uważana obecnie za klasyczną i jednocześnie najczęściej przytaczana przez wielu znanych badaczy definicja ontologii w odniesieniu do informatyki została zaproponowana przez Grubera w 1993 roku. Brzmi ona następująco: „Ontologia jest »formalną specyfikacją wspólnej warstwy pojęciowej«" [3, 4, 5].

Koncepcja ontologii A. Meadche [6] opiera się na definicji jej struktury i leksykonie. Podstawowym jej elementem są pojęcia (odpowiadające klasom), opisujące pewną grupę obiektów o podobnych cechach. Struktura ontologii definiująca pojęcia i relacje między nimi ma postać [6]:

$$O = \{C, R, Hc, Rel, A\} \tag{1}$$

gdzie:

- C – zbiór wszystkich pojęć wykorzystanych w modelu,
- R – zbiór nietaksonomicznych relacji (zwanych właściwościami, slotami, rolami), definiowanych jako nazwane połączenie między pojęciami,
- Hc – zbiór taksonomicznych relacji pomiędzy konceptami,
- Rel – zdefiniowane nietaksonomiczne relacje pomiędzy pojęciami,
- A – zbiór aksjomatów.

Leksykon zawiera interpretacje pojęć i relacji występujących miedzy nimi:

$$L = \{Lc, Lr, F, G\} \tag{2}$$

gdzie:

- Lc – definicje leksykonu dla zbioru pojęć,
- Lr – definicje leksykonu dla zbioru relacji,

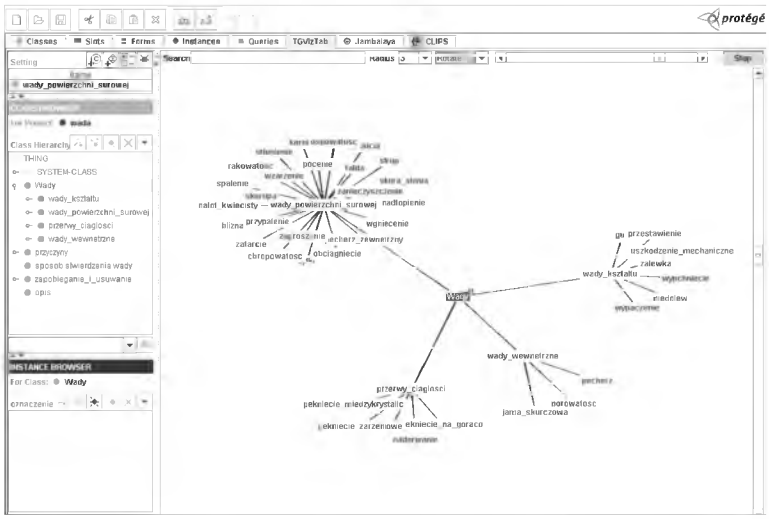
F – referencje dla pojęć,
G – referencje dla relacji.

Na potrzeby diagnostyki wad odlewów zbudowano ontologię, wykorzystując edytor Protégé 2000. Program ten jest bezpłatnym narzędziem rozprowadzanym na zasadzie licencji *open source*. Jest to aplikacja wspomagająca tworzenie baz wiedzy, w tym, między innymi, baz służących do edycji ontologii i pozyskiwania wiedzy od ekspertów.

Platforma Protégé umożliwia zastosowanie dwóch form reprezentacji wiedzy przez dwie autonomiczne aplikacje Protégé-Frames i Protégé-OWL. Ontologie zbudowane za pomocą aplikacji Protégé mogą być eksportowane do wielu różnych formatów, w tym między innymi RDF(S), OWL i XML Schema.

Aplikacja Protégé jest aplikacją w języku Java, a jedną z jej wielu zalet stanowi możliwość rozszerzania jej funkcjonalności przez doinstalowanie pluginów. Można także rozszerzać możliwości aplikacji samodzielnie przez wykonywanie wtyczek do niej, a także przez budowanie własnych ontologii [7].

Najczęściej model wiedzy służący do budowy ontologii oparty jest na ramach, które stanowią podstawę konstrukcji bloków ontologii. Model wiedzy może wykorzystywać kilka typów ram: klasy, sloty i instancje. Klasy definiują nową ontologię. Instancje stanowią elementy tego zbioru, czyli klas. Klasy mogą mieć strukturę hierarchiczną. Sloty dają możliwość definicji zarówno poszczególnych właściwości danych klas, jak i relacji zachodzących miedzy nimi. Ontologie wraz z instancjami klas – pojęć – tworzą łącznie bazę wiedzy.



Rysunek 2. Hierarchia klas „wady" pokazana za pomocą pluginu TGVizTab

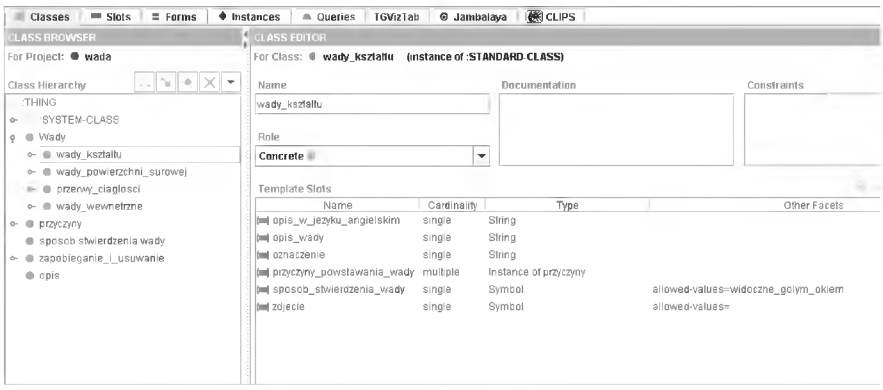
Zbudowano system ontologiczny, w którym główną klasą są „wady". Klasa ta składa się z czterech podstawowych podklas:

- „wady_kształtu",
- „wady_powierzchni",
- „wady_wewnętrzne",
- „przerwy_ciągłości".

Do każdej z tych klas zostały przyporządkowane wady. Głównym kluczem ich przyporządkowania była przyczyna powstania. Rysunek 2 przedstawia podział klasy „wady" na wymienione podklasy oraz przyporządkowane każdej z nich wady.

Każda z wad scharakteryzowana została poprzez sloty:

- nazwa w języku angielskim,
- opis wady w języku polskim,
- oznaczenie wady, czyli symbol, jakim została zdefiniowana (np. W-101 oznacza uszkodzenie mechaniczne, oznaczenie to pochodzi z polskiej normy wad odlewów) [8],
- przyczyna powstania wady (opis możliwych przyczyn, jakie mogą wywołać powstanie danej wady),
- sposób stwierdzenia wady (wskazówki co do tego, w jaki sposób stwierdzono jej zaistnienie – czy można wadę zdiagnozować gołym okiem czy trzeba wykonać specjalistyczne badania),
- zdjęcie wady jest to slot, w którym można zobaczyć zdjęcie przykładowej wady.



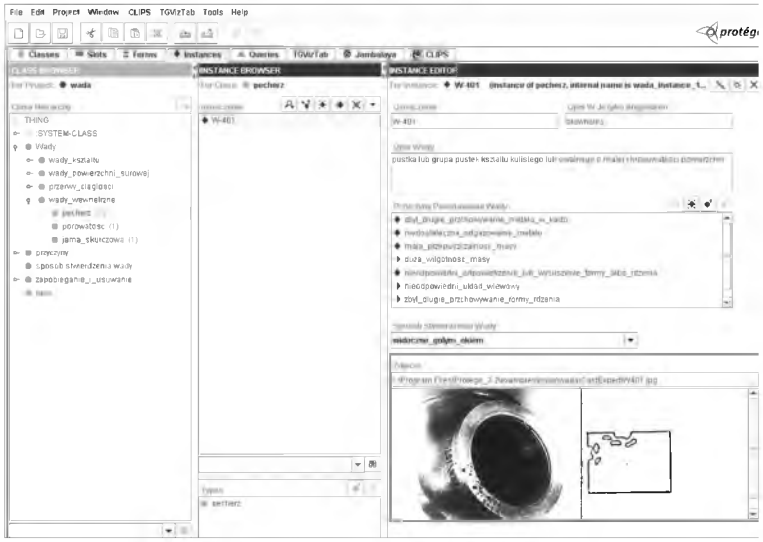
Rysunek 3. Cechy podklasy „wady_kształtu"

Na rysunku 3 przedstawiono cechy podklasy „wady_kształtu", gdzie wszystkie sloty dziedziczone są z klasy „wady". Slot „przyczyny_powstawania_wady"

jest typu Instance, czyli jest to typ obiektowy klasy „przyczyny". Wypełnienie slotów konkretnymi wartościami tworzy instancje danej klasy. Przykład wypełnienia slotów dla wady „pęcherz" znajduje się na rysunku 4. W odpowiednich slotach została opisana charakterystyka tej wady.

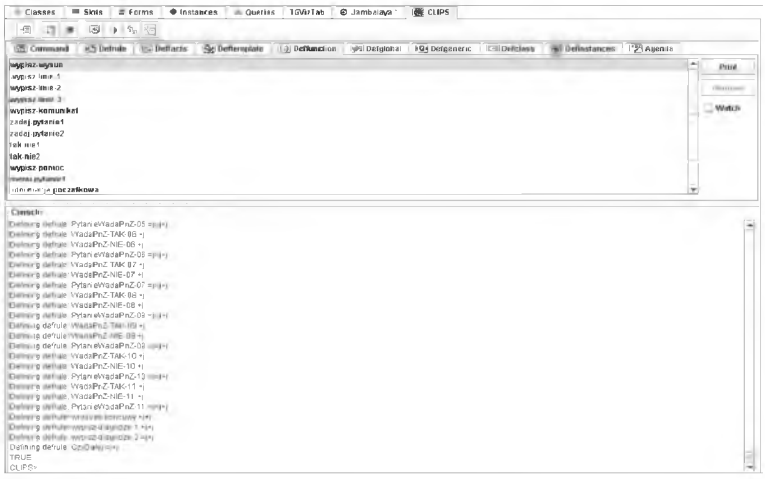
Pakiet Protégé-2000 dostarcza także narzędzia pozwalającego na przeglądanie oraz wydobywanie informacji ze stworzonej bazy wiedzy. Na rysunku 5 przedstawiono możliwość posortowania wad w zależności od przyczyny ich powstania. W przykładzie tym zostały wyświetlone wady, dla których w slocie „przyczyna wystąpienia wady" znajduje się stwierdzenie, że jest nią „niedostateczne_odgazowanie". Umożliwia to stworzenie reguły dla systemu ekspertowego, w której użytkownik może uzyskać informację o tym, jakie wady mogą powstać w odlewie w przypadku zaistnienia przyczyny „niedostateczne_odgazowanie":

If *niedostateczne odgazowanie* **then** *pęcherz zewnętrzny* **or** *nakłucia* **or** *pęcherz* **or** *porowatość.*

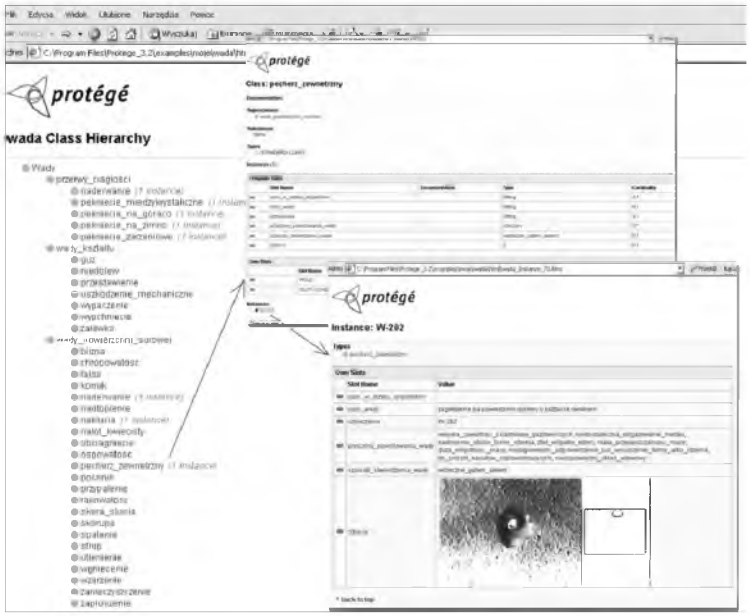


Rysunek 4. Instancja klasy W-401

Rysunek 5. Spis wad, które mogą powstać w wyniku przyczyny „niedostateczne_odgazowanie"



Rysunek 6. Przykład testowego systemu ekspertowego



Rysunek 7. Dokument w standardzie HTML

System wyposażony w dużą ilość prawdziwej i dokładnej wiedzy z dziedziny wad wyrobów metalowych będzie podstawą do zbudowania regułowego systemu ekspertowego. Tę możliwość daje plugin CLIPS programu Protégé-2000.

Edytor Protégé umożliwia również wygenerowanie dokumentacji w standardzie HTML – każdy element tworzonej ontologii można opisać i automatycznie dołączyć do dokumentacji całego projektu.

3. Podsumowanie

W artykule przedstawiono sposób postępowania prowadzący do wytworzenia systemu wspomagającego zarządzanie wiedzą. Proponowana metodyka dostarcza konkretnego materiału dla inżynierów planujących proces technologiczny wytwarzania wyrobów metalowych. Inżynier procesu czy kierownik projektu, planując proces techniczny, dostaje wsparcie w postaci gotowych opisów klas, ról i relacji między nimi. W ramach kontynuacji prac przewiduje się uzupełnienie brakujących fragmentów wiedzy. Skonstruowana ontologia wad odlewów zostanie wykorzystana jako istotny element systemu diagnostycznego, ułatwiający technologom zrozumienie związków przyczynowo-skutkowych w procesie technologicznym, a w konsekwencji – poprawę efektywności jego realizacji.

Bibliografia

[1] Kluska-Nawarecka S., *Metody komputerowe wspomagania diagnostyki wad odlewów*, Krakow 1999.

[2] Mrzygłód B., Kluska-Nawarecka S., *Collecting and Formalization of Knowledge about Surface Faults of Metal Products*, „Archives of Foundry" 2005, vol. 5, No. 17, s. 175–182 (in Polish).

[3] Gruber T., *A Translation Approach to Portable Ontology Specifications*, „Knowledge Acquisition" 1993, vol. 5, s. 199–220.

[4] Kluska-Nawarecka S., Ciszewski S., *Document Driven Ontological Engineering with Applications in Casting Defects Diagnostic*, 2004, vol. 4, No. 1–2, s. 56–64 (in Polish).

[5] Podsiadły-Marczykowska T., Guzik A., *Mammographic Ontology – Structure of the Model, Definitions and Instances of Concepts Bio-algorithms and Med-systems*, „Journal Edited by Medical College – Jagiellonian University" 2005, vol. 1, No. 1/2, s. 247–252 (in Polish).

[6] Maedche A., *Ontology Learning for the Semantic Web*, „Intelligent Systems, IEEE" 2001, vol. 16, No. 2, s. 72–79.

[7] http://protege.stanford.edu.

[8] Polski Komitet Normalizacji Miar i Jakości, PN-85/H-83105, Odlewy: Podział i terminologia wad, Dz. Norm i Miar 1, pozycja 2, 1986.
[9] Kluska-Nawarecka S., Tybulczuk J., Kisiel-Dorohinicki M., Połcik H., Nawarecki E., *Distributed Information System for Foundry Industry with Application of Multi-agent*, „Archives of Foundry" 2006, vol. 6, No. 18, s. 45–52 (in Polish).
[10] Kluska-Nawarecka S., Ciszewski S., Adrian A., *Od eksperta do bazy wiedzy. Inteligentne metody komputerowe dla nauki, technologii i gospodarki*, Warszawa 2004.
[11] Kluska-Nawarecka S., Dobrowolski G., Marcjan R., Nawarecki E., *OntoGRator An Intelligent Access to Heterogeneous Knowledge Sources about Casting Technology*, „Computer Methods in Materials Science" 2007, vol. 7, No. 2, s. 324–328 (in Polish).

Dorota Wilk-Kołodziejczyk
Agnieszka Smolarek-Grzyb
Krzysztof Regulski

Diagnostyczne systemy ekspertowe z wykorzystaniem logiki wiarygodnego rozumowania

Artykuł przedstawia założenia rozszerzenia systemu ekspertowego CastExpert, służącego do diagnostyki wad wyrobów odlewniczych. Na przykładzie ekspertyzy wady o nazwie „pęknięcie na zimno" zostanie zaprezentowany mechanizm wnioskowania w przód, oparty na zaimplementowanej bazie reguł. W implementacji wykorzystano ranking przyczyn powstania wady, stworzony z wykorzystaniem logiki wiarygodnego rozumowania. W rezultacie w artykule przedstawiono koncepcję algorytmu prowadzenia przez system dialogu z użytkownikiem. W dialogu tym kolejność zadawania pytań jest uzależniona od rangi, jaką otrzymają określone przyczyny występowania wad.

1. Systemy ekspertowe w diagnostyce

System ekspertowy to program wykorzystujący wiedzę i procedury rozumowania do wspomagania rozwiązywania problemów na tyle trudnych, że do ich pokonania potrzebna jest wiedza eksperta [1]. Stanowi on jedną z gałęzi stosowanej sztucznej inteligencji, a jego podstawowa idea polega na przeniesieniu wiedzy eksperta do programu komputerowego, wyposażonego w bazę wiedzy, konkretne reguły wnioskowania i język komunikacji z użytkownikiem lub interfejs graficzny na taką komunikację pozwalający. Cała wiedza zgromadzona w systemie może być wykorzystywana wielokrotnie przez wielu użytkowników, zwracających się do komputera za każdym razem, kiedy potrzebują rady. Komputer powinien zwracać najlepsze rozwiązanie problemu i jeśli to konieczne, tłumaczyć

logikę, na podstawie której powstały wnioski wyjściowe. Główne zadania stawiane przed diagnostycznym systemem ekspertowym to: ocena danego środowiska na podstawie obserwacji, wykrywanie wad, podawanie sposobu postępowania w przypadku nieprawidłowego funkcjonowania danego obiektu [2].

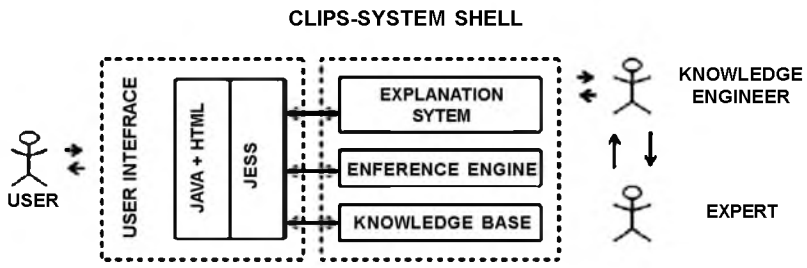
Klasycznym językiem używanym przy tworzeniu systemów eksperckich jest Prolog. Obecnie, zamiast tworzyć systemy ekspertowe od podstaw, używa się ich gotowych szkieletów (*expert system shell*) z zaimplementowanym mechanizmem wnioskowania. Szkielet taki to właściwie system ekspertowy z pustą bazą wiedzy [3].

Jednym ze szkieletów wykorzystanych w systemie CastExpert jest CLIPS (*C Language Integrated Production System*), stworzony w 1985 roku przez Johnson Space Center (NASA). Od tamtego czasu jest rozwijany, obecnie obowiązuje wersja 6.24. Zapewnia on środowisko programistyczne do tworzenia systemów ekspertowych oraz zawiera narzędzia do tworzenia systemów bazujących na rozumowaniu regułowym i reprezentacji wiedzy. Daje także programistom możliwość rozbudowy [4]. Innym przykładem jest JESS (*Java Expert System Shell*), stworzony w 1995 roku przez Sandia National Laboratories (organizacja rządowa USA). Początkowo miał być tylko interpreterem CLIPS-a pisanym w języku Java, jednak obecnie stanowi już niezależny *shell* [5].

2. System CastExpert

CastExpert jest systemem ekspertowym, który na podstawie szczegółowej wiedzy może wyciągać wnioski, działając w sposób zbliżony do rozumowania człowieka. Opiera się na technice symbolicznego przetwarzania informacji, a stosowaną w nim formą zapisu wiedzy są reguły [6]. Prototypowa wersja systemu CastExpert została uruchomiona na serwerach Instytutu Odlewnictwa w Krakowie. Od tego czasu trwają prace badawcze związane z rozwojem i integracją z innymi modułami rozproszonego systemu informacyjnego INFOCAST oraz systemów zarządzania wiedzą. Cenną zaletą systemu są jego małe wymagania sprzętowe – do obsługi wymagana jest jedynie przeglądarka obsługująca aplety Javy. Sam mechanizm wnioskujący i podsystem współpracy z użytkownikiem umieszczone są na serwerze, co daje tym samym możliwość pracy zdalnej przez sieć Internet (rysunek 1).

Mechanizm wnioskujący został oparty na szkielecie systemów ekspertowych (*expert system shell*) z zaimplementowanym mechanizmem wnioskowania, jakim jest CLIPS, a także wspomagającym go interpreterze Javy – JESS. Taka konstrukcja zapewnia możliwość prostej obsługi przez przeglądarkę oraz rozbudowy i współpracy z innymi modułami. CLIPS wykorzystuje mechanizm wnios-



Rysunek 1. Schemat architektury systemu CastExpert

kowania w przód (*forward chaining*). W kolejnych krokach następują poszczególne czynności [7]:

1. Sprawdzenie, czy przesłanki którejś z reguł są faktami zaobserwowanymi przez użytkownika; jeżeli tak, to taka reguła jest uaktywniana.
2. Konkluzja tak wybranej reguły jest wprowadzana jako nowy fakt do bazy wiedzy.

Powrót do punktu 1 do momentu, kiedy wśród wygenerowanych faktów znajdzie się postawiony cel (diagnoza) lub gdy nie ma w bazie wiedzy więcej reguł do uaktywnienia.

3. Przerwy ciągłości – pęknięcie na zimno

Pęknięcie na zimno jest jedną z najtrudniej diagnozowalnych wad odlewów. Wynika to stąd, że pęknięcia są niepowtarzalne. Każdorazowo może zaistnieć inna przyczyna pękania, nawet w przypadku odlewów z tego samego wytopu.

Wada powstaje wówczas, gdy naprężenia wewnętrzne wywołane nierównomierną zmianą temperatury odlewu przekraczają wytrzymałość materiału bądź też gdy sumują się z nimi dodatkowe naprężenia wynikające z innych przyczyn, np. mechanicznego hamowania skurczu, stosunkowo nawet słabych uderzeń, przemian fazowych. Na skłonność do pęknięć na zimno znaczny wpływ wywierają skład chemiczny i zanieczyszczenia stali [8]. Można jednak wymienić dość dużą grupę przyczyn powstawania pęknięć:

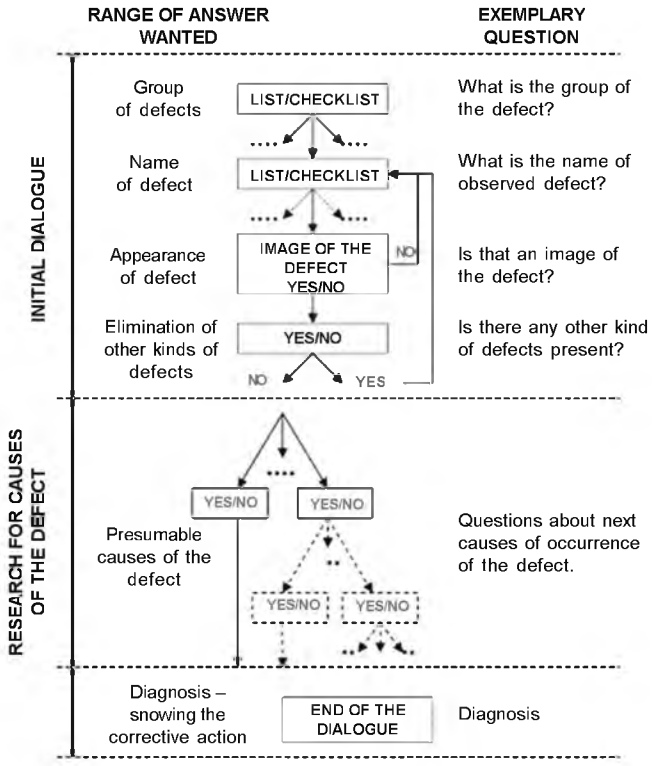
1. Nadmierne różnice grubości ścianek.
2. Zbyt małe, zbyt grube lub za rzadko rozstawiane żeberka wzmacniające odlew albo ich brak.
3. Niedostateczna podatność masy, nadmiernie mocne ubicie formy lub rdzenia.
4. Wilgotność masy formierskiej.

5. Rozmieszczenie i przekroje wlewów doprowadzających uniemożliwiające równoczesne krzepnięcie.
6. Ukształtowanie wlewów, nadlewów lub uzbrojenia formy powodujące hamowanie skurczu.
7. Niedostateczna wielkość lub brak ochładzalnika zewnętrznego lub ochładzalników wewnętrznych.
8. Nieprawidłowo zastosowany sposób odlewania na sucho.
9. Niekorzystne dla danego odlewu zastosowanie kwaśnego procesu topienia.
10. Zbyt niska zawartość manganu.
11. Zbyt wysoka zawartość fosforu, wodoru, azotu.
12. Stopień odtlenienia metalu.
13. Zbyt wysoka zawartość siarki i innych wtrąceń niemetalicznych.
14. Za wysoka temperatura spustu i/lub zalewania.
15. Zbyt wczesne wybicie odlewu i/lub nieodpowiednie wybijanie.
16. Zbyt szybkie i/lub nierównomierne studzenie odlewu.
17. Nieostrożny transport i/lub składowanie.
18. Niewłaściwe usuwanie wlewów i nadlewów.
19. Niewłaściwy proces obróbki cieplnej i/lub spawania.

Jak widać, lista przyczyn jest długa, ponadto w wielu przypadkach do wady może prowadzić równoczesne wystąpienie kilku z nich, co utrudnia proces diagnostyczny. System CastExpert ma wspierać użytkownika w odnalezieniu przyczyny przez zadawanie mu pytań związanych z procesem produkcyjnym, a następnie podanie sposobu postępowania w celu wyeliminowania wady.

4. Schemat procedury diagnostycznej

Zadaniem systemu ekspertowego jest przeprowadzenie diagnozy stanu zastanego przez wyciąganie wniosków przy użyciu zaprogramowanych reguł w sposób zbliżony do ludzkiego rozumowania. Reguły takie można zapisać w sposób symboliczny jako: *IF <przesłanka> THEN <konkluzja>*. CastExpert wykorzystuje do wnioskowania również fakty (przesłanki), których dostarcza użytkownik. Pierwszym etapem procesu diagnostycznego jest tzw. dialog wstępny (rysunek 2), który umożliwia użytkownikowi określenie tego, jaki stop należy brać pod uwagę i jaki rodzaj wady wystąpił w odlewie. Użytkownik ma też możliwość porównania wyglądu wady z bazą przypadków zebranych w systemie, aby na tej podstawie móc wskazać rodzaj wady.



Rysunek 2. Schemat diagnostyki w systemie CastExpert
Źródło: [6].

Część zasadnicza procesu diagnostycznego to poszukiwanie przyczyny wystąpienia wady (rysunek 2). System prowadzi dialog z użytkownikiem, zadając mu kolejne pytania, dzięki którym możliwe jest wyeliminowanie tych potencjalnych przyczyn, które na pewno w diagnozowanym procesie nie miały miejsca.

Przykładowa reguła dotycząca przyczyny powstawania pęknięć na zimno zapisana w języku CLIPS ma następującą postać:

```
(defrule conclusion-a19-right-cast-construction
(not (defect yes))
(defect-kind ?)
(a19-right-cast-construction no) =>
(assert (defect yes))
(assert (diagnosis defect-r140)))
(defrule check-a19-right-cast-construction
```

```
(declare (salience -20))
(not (defect yes))
(diagnosis)
(defect-kind ?)
(not (a19-right-cast-construction ?)) =>
(bind ?answer (question „Czy konstrukcja jest prawidłowa?”))
(assert (a19-right-cast-construction ?answer)))
```

Odpowiada ona za zadanie użytkownikowi podczas prowadzonego dialogu pytania o jedną z przyczyn powstawania wady oznaczonej jako W-302 (pęknięcie na zimno/*cold crack*). Na pytanie: „Czy konstrukcja odlewu jest poprawna?” użytkownik odpowiada „Tak/Nie/Nie wiem”. Odpowiedź „Nie” zaowocuje podaniem diagnozy:

```
(defrule print-diagnosis-r140
(defect yes)
(diagnosis defect-r140) => (diag (create$

"1. Analiza konstrukcji odlewu."
"2. Unikanie nadmiernych różnic w grubości ścian odlewu."
```

W podobny sposób przebiega dialog w przypadku pozostałych zdiagnozowanych we wstępnym etapie wad. Po odnalezieniu przyczyny system podaje sposób działania zmierzający do usunięcia wady i przywrócenia prawidłowości procesu.

5. Algorytm prowadzenia dialogu zależny od rangi przyczyn

System prowadzi dialog z użytkownikiem, zadając mu pytania, na które odpowiedź może ujawnić przyczynę powstania wady. Liczba pytań zależna jest od momentu odkrycia zaistniałej przyczyny. Jeśli już w pierwszym pytaniu uda się zdiagnozować przyczynę, dialog zostaje zakończony. Jeżeli przyczyna nie zostanie ujawniona podczas dialogu, oznacza to, że zostały zadane wszystkie pytania zaimplementowane w systemie, czyli w bazie nie ma odpowiednich reguł, które pozwoliłyby określić przyczynę danej wady. Można sobie zatem wyobrazić sytuację, w której użytkownik zmuszony jest odpowiedzieć na wiele pytań, zanim uzyska satysfakcjonujący wynik. Warto też zauważyć, że jeśli zaistniało kilka przyczyn na raz, użytkownik po każdorazowym usunięciu przyczyny powtarza

cały dialog, aby odkryć kolejną nieprawidłowość w procesie odlewania. Może być to żmudny proces odpytywania, który zmniejsza użyteczność systemu.

W obecnym stanie systemu kolejność zadawania pytań, a w związku z tym wywoływania odpowiednich reguł z bazy wiedzy zależna jest od czynników pozamerytorycznych, a mianowicie od kolejności wprowadzania do systemu, a także od autorytarnej decyzji programisty, a nie technologa. Istnieje jednak istotna cecha nadawana regułom w trakcie implementacji. Jest nią ich „ważność" (*salience*). Parametr SALIENCE określa priorytet reguły względem innych reguł w bazie. Domyślnie parametr ten posiada wartość równą zeru, nadanie mu wartości dodatniej powoduje przyporządkowanie mu wyższego priorytetu, analogicznie: im bardziej ujemny parametr, tym niższy priorytet. Reguły o najwyższym priorytecie uruchamiane są najwcześniej (przed innymi regułami).

W celu zwiększenia użyteczności systemu i tym samym odczuwanej przez użytkownika jakości komunikacji z systemem należałoby tak ustawić kolejność zadawania pytań, aby te przyczyny, które najczęściej powodują daną wadę, zostały wyeliminowane już w pierwszych pytaniach, co w większości przypadków znacznie skróci dialog. W tym celu należy posłużyć się rankingiem przyczyn wykonanym przy użyciu logiki wiarygodnego rozumowania.

Parametr SALIENCE powinien pozwolić określać priorytet reguły w sposób dynamiczny, zależnie od rangi uzyskanej w rankingu przyczyn powstawania wady. I tak dla pęknięć na zimno kolejność zadawania pytań zależna będzie od najczęściej występujących grup przyczyn:

1. Duża różnica temperatur w stygnącym odlewie, powodująca zmianę naprężeń, a w konsekwencji pęknięcia; dzieje się tak wtedy, gdy różnice w grubości ścian odlewu są duże lub niewłaściwie umieszczony jest układ wlewowy powodujący duże gradienty temperatur.
2. Źle dobrane warunki chłodzenia.
3. Zawartość fosforu i innych niepotrzebnych gazów.

Wprowadzając informacje o danej wadzie, korzystamy z kilku źródeł informacji. Źródła te to normy, atlasy, informacje od ekspertów [5, 6, 7, 8, 9]. W każdym z nich wada opisana jest za pomocą różnych parametrów. Jeden z parametrów służących do opisu to przyczyna wystąpienia wady. Uwzględniając te same przyczyny wystąpienia wady wskazywane w różnych opisach pęknięcia na zimno, możemy doprowadzić do automatycznego ustalania parametru SALIENCE. Im więcej źródeł będzie wymieniać daną przyczynę, tym wyższy będzie ten parametr.

Do sporządzenia odpowiedniego rankingu użyjemy sposobu wnioskowania zaczerpniętego z logiki wiarygodnego rozumowania (LPR), w którym następuje transformacja wartości w stwierdzeniach. Zapis ogólny tej transformacji wygląda następująco:

D(a) = {R,.... }: γ_1 φ, μ_r
R' SIM R in CX (d, D(d)); γ_2, σ
D(d)-A(d):γ_3,α
a SPEC A: γ_4
---
d(a) = {R^1, ...}: $\gamma = f(\gamma_1 \varphi, \mu_r \gamma_2, \sigma, \gamma_3, \alpha, \gamma_4)$

Transformacja ta zastosowana do naszej wady będzie miała następującą postać:

Przyczyna(pęknięcie na zimno) = {R^P,....}
R^{cz} SIM R^P in CX (przyczyna, Przyczyna powstania wady(przyczyna))
konstrukcjaOdlewu(przyczyna)-przyczynaWady(przyczyna)
Pęknięcie na zimno SPEC Konstrukcja odlewu:
---
d(a) = {R^{cz}, ...})

R^P – przyczyna występująca w normie polskiej,
R^{cz} – przyczyna występująca w opracowaniu czeskim.

Z wnioskowania tego wynika, że przyczyna wystąpienia wady pęknięcie na zimno występująca w jednym z opracowań polskich może zostać także odnaleziona w innym opracowaniu, np. czeskim. Nie jest to jednoznaczne, gdyż jak wynika z analizy zebranego materiału, w każdym ze źródeł przyczyny są inaczej sformułowane i nie zawsze wszystkie wymienione.

6. Podsumowanie

Prezentowany projekt rozwiązania pozwala użytkownikowi na sprawne komunikowanie się z systemem. Niektóre z wad wyrobów odlewniczych mogą mieć bardzo wiele przyczyn i dojście do tej rzeczywistej bywa niezwykle trudne. Dlatego ważne ze względów użytkowych jest zastosowanie rankingów przyczyn.

Bibliografia

[1] Giarratano J.C., Riley G., *Expert Systems. Principles and Programming*, Thomson Course Technology, 2004.
[2] Waterman D.A., *A Guide to Expert Systems*, Addison-Wesley, 1986.
[3] Mulawka J.J., *Systemy ekspertowe*, Warszawa 1996.
[4] CLIPS, http://www.ghg.net/clips/CLIPS.html.
[5] JESS, http://www.jessrules.com/jess.

[6] Kluska-Nawarecka S., Dobrowolski G., Marcjan R., Nawarecki E., *Od pasywnych do aktywnych źródeł danych i wiedzy. Zdecentralizowany system informacyjno-decyzyjny dla wspomagania technologii odlewniczej*, Kraków 2002.
[7] Niederliński A., *Regułowo-Modelowe Systemy Ekspertowe*, Bielsko-Biała 2006.
[8] *Systematyka wad odlewów staliwnych*, Warszawa 1954.
[9] Collins A., Michalski R.S., *The Logic of Plausible Reasoning: A Core Theory*, „Cognitive Science" 1989, vol. 13, No. 1, s. 1–49.
[10] Collins A., *Fragments of a Theory of Human Plausible reasoning, Association for Computational Linguistics*, University of Illinois, 1978.
[11] Michalski R.S., *A Theory and Methodology of Inductive Learning*, [w:] R.S. Michalski, J.G. Carbonell, T.M. Mitchell (eds.), *Machine Learning: An Artificial Intelligence Approach*, Palo Alto 1983.
[12] Bohem-Davis D., Dontas K., Michalski R.S., *A Validation and Exploration of the Collins–Michalski Theory of Plausible Reasoning. Reports of the Machine Learn ing and Inference Laboratory*, George Mason University, 1990.
[13] Kluska-Nawarecka S., Wilk-Kołodziejczyk D., *Integration of Heterogeneous Knowledge Components Using Logic of Plausible Reasoning, Conference on: Intelligent Information Systems*, Poznań 2005.
[14] Podział i terminologia wad, Polski Komitet Normalizacji Miar i Jakości, PN-85/H-83105, Dz. Norm i Miar 1, pozycja 2, 1986.
[15] Hèron G., Mascaré C., Blanc G., *Recherche de la qualité des pièces de fonderie*, Paris 1986.
[16] Elbel T., Havliček F., Jelinek P., Leviček P., Rous J., Stransky K., *Vady odlitků ze slitin železa (klasifikace, přičiny a prevence)*, Brno 1992.
[17] Baler J., Köppen M., *Podręcznik wad odlewniczych, wady związane z masami formierskimi i zapobieganie ich powstawaniu*, Maral 1994.
[18] Hasse S., *Guß- und Gefügefehler, Erkenung, Deutung, und Vermeidung von Guss und Gefügefehlerbei der Erzeugung von gegossenen komponenten*, Berlin 2003.

Romuald Wit

Generatory liczb (pseudo-)losowych

1. Wstęp

Jeszcze na sam koniec XIX wieku Lord Rayleigh pokazał, że wartości przybliżonego rozwiązania parabolicznego równania różniczkowego można uzyskać w procesie błądzenia losowego. Z końcem lat dwudziestych ubiegłego stulecia pojawiły się prace Couranta, Kołmogorowa i Pietrowa pokazujące związek dwuwymiarowego błądzenia przypadkowego z rozwiązaniami zagadnienia Dirichleta. Stało się jasne, że rozwiązania dotyczące problemów stochastycznych mogą mieć swoje odniesienie do przybliżonych rozwiązań odpowiednich zagadnień, niekoniecznie stochastycznej proweniencji. W następnej zaś kolejności zaproponowano swoiste odwrócenie ról: zaczęto na większą skalę stosować metody stochastyczne tam, gdzie metody różniczkowe (np. równania transportu) lub różnicowe okazały się niewydolne.

Wprowadzony do obiegu naukowego przez Nicolasa Metropolisa i Stanisława Ulama, chyba nie bez pewnego przymrużenia oka, termin „metody Monte Carlo" [1] stał się dobrze rozpoznawalnym znakiem firmowym metod obliczeniowych szeroko stosowanych w różnych dziedzinach nauki. Metody te dostarczają przybliżonych rozwiązań rozmaitych matematycznych problemów za pomocą statystycznych obliczeń komputerowych. Używają tych metod fizycy, biolodzy, chemicy, medycy, technicy, wykorzystuje się je i w inżynierii finansowej [2]. Mają one zastosowanie na ogół wszędzie tam, gdzie mamy do czynienia z modelowaniem jakiejś części naszej rzeczywistości. Modelowanie chociażby transportu neutronów (projekt „Manhattan") było tego dobrym przykładem.

Kluczową rolę w metodach Monte Carlo odgrywają (programowe) generatory *liczb losowych*. Jest wprawdzie pewną ironią losu, że tak deterministyczne urządzenia jak komputery służą również do generowania liczb o charakterze (prawie...) losowym, ale nad tym nikt się już chyba w tej chwili nie zastanawia[1].

[1] Świadomość tego faktu najlepiej oddaje cytat sprzed 70 lat z pracy J. von Neumanna, autora jednego z najstarszych znanych generatorów liczb losowych, tzw. generatora kwadratowego: „Any one who considers arithmetical methods of producing random digits, is of course in a state of sin".

To po prostu działa. Tablice liczb losowych, które można było znaleźć w niektórych starszych podręcznikach statystyki, mogłyby być zastąpione np. przez płyty CD zawierające olbrzymią ilość „dobrze obliczonych" liczb losowych (lub też mogłyby być schowane w odpowiednich zbiorach „na sieci"), aczkolwiek większy komfort psychiczny daje użytkownikowi, rzecz jasna, posiadanie „własnego" generatora.

Kluczem do zrozumienia dalszych wywodów są słowa „dobrze obliczonych". Sprawa nie jest trywialna – jeśli modele, których używamy do opisu wycinka naszej rzeczywistości, mają mieć jakikolwiek sens, to:

- one same muszą mieć poprawną wewnętrzną strukturę i być w jakiejś mierze adekwatne do opisywanej rzeczywistości,
- nie mogą korzystać z wybrakowanych narzędzi, które będą nam „kreować rzeczywistość" (w dodatku w sposób niekontrolowany...).

Pierwszy punkt jest dosyć oczywisty, drugi – trochę mniej. W dość powszechnym odczuciu wszystko, co przychodzi „z komputera", jest do zagospodarowania. Wystarczy jednak przyjrzeć się bezkrytycznemu korzystaniu z zasobów Internetu, wystarczy przypomnieć sobie niechlubną historię generatora liczb losowych RANDU firmy IBM, by wykazać w tej sprawie nieco więcej ostrożności[2]. W niektórych eksperymentach numerycznych mamy do wygenerowania dziesiątki *milionów* liczb losowych odpowiedniej jakości.

Trudno się więc dziwić, że przy teoretycznym omawianiu własności generatorów liczb losowych głęboko zaangażowana jest teoria liczb, że produkty generatorów przechodzą naprawdę trudne, bardzo różnorodne testy statystyczne (i często ich nie zdają! – por. [3] czy też pakiet DIEHARD G. Marsaglii z 1996 roku), że ich optymalnemu kodowaniu (C/C++, Fortran, asemblery itp., por. [4, 5]) poświęca się wiele uwagi. W końcu każda licząca się w tej chwili biblioteka programów matematycznych zawiera jakiś łatwo dostępny generator liczb losowych.

Dość długo dwiema głównymi zmorami generatorów liczb losowych były:

- periodyczność wyprodukowanych liczb losowych (krótkie okresy); tytułem wyjaśnienia: okres rzędu 10^{18} już uważa się za krótki...
- korelacje między wyprodukowanymi liczbami losowymi.

Pierwszy z wymienionych wyżej problemów chyba znalazł, jak na razie, swoje pozytywne rozwiązanie. Szacuje się, że obecnie zalecane do użytku generatory mają okresy rzędu 10^{171}. Praktycznie nikt tego chyba nie sprawdzi... Pozostaje jednak problemem, czy te generatory dostarczają liczb losowych o wymaganych własnościach statystycznych i czy robią to dostatecznie efektywnie.

[2] Kasyna też potrzebują liczb losowych...

2. Generatory

Niewątpliwie „na rynku" przez bardzo długi czas najbardziej popularne były generatory oparte na konstrukcji Lehmera [6] z końca lat czterdziestych ubiegłego wieku:

$$x_{n+1} = (a * x_n + c) \quad mod \quad M, \quad n = 0, 1, 2, \ldots$$

Jest to liniowy, kongruetny generator mieszany (występuje w nim dodawanie i mnożenie). Jakość produkowanych przez niego ciągów liczb $\{x_n\}$ zależy od wyboru występujących w nim stałych: a, c oraz M (trochę mniej od x_0; jest to początek ciągu x_n i nosi on nazwę *ziarna*, *zasiewu*; ang. *seed point*). Wybór wielkości a, c oraz M odbywa się według pewnego zespołu wskazówek [7]. Korzystne ich realizacje znajdzie Czytelnik np. w monografii [8] w postaci tabel z uwagami dotyczącymi zakresu używanych liczb. Nie od rzeczy będzie jednak wspomnieć, że proste generatory tego typu kwalifikuje się dziś po prostu jako „quick and dirty".

Wszystkie występujące tu liczby są liczbami naturalnymi. Przerzucenie ciągu liczb naturalnych $\{x_n\}$ na ciąg liczb rzeczywistych z przedziału $[0,1]$ nie stanowi żadnego problemu. A potem możemy już przejść, w razie potrzeby, do budowy generatorów o zupełnie innych rozkładach (np. wykładniczym, normalnym, Cauchy'ego, Lévy'ego).

Na nieco innej zasadzie były budowane generatory typu Fibonacciego [9] w postaci:

$$x_n = (x_{n-r} \pm x_{n-q} \pm c) \quad \text{mod} \quad M,$$

gdzie przykładowe wartości stałych to $M = 2^{24}$, $r = 24$, $s = 10$, a stała c stanowi bit przeniesienia (0 lub 1). Oznacza to, że do inicjalizacji pracy tego generatora potrzebujemy (w zasadzie) dowolnych 24 liczb (ich wybór nie może być całkiem trywialny, np. nie mogą to być same zera). Liczby losowe wygenerowane według powyższego przepisu nie przechodziły jednak np. testu odstępów (ang. *gap test*).

Nowe spojrzenie na działanie generatorów liczb losowych zaproponował Lüscher [10]. Potraktował on generator liczb losowych jako pewien *układ dynamiczny*, odwzorowujący jeden stan generatora na następny.

W r-wymiarowej przestrzeni jednostkowy hipersześcian jest zbiorem wszystkich wektorów

$$\vec{v} = (v_0, v_1, \ldots, v_{r-1})^T$$

o składowych $\upsilon_i \in [0, 1]$. Identyfikując przeciwległe brzegi tego hiperześcianu, otrzymujemy w r-wymiarowej przestrzeni torus T_r, którego podzbiorem jest dyskretny torus T_r^d składający się ze wszystkich wektorów o składowych:

$$\upsilon_l = n_l/M, \quad n_l = 0, 1, 2, \ldots, M-1.$$

Wektory

$$\vec{\upsilon}(t) = (x_n, x_{n+1}, \ldots, x_{n+r-1})^T/M, \quad n = rt$$

określają pewien punkt na torusie T_r^d, poruszający się z dyskretnym, jednostkowym wzrostem „czasu". Możemy zatem napisać:

$$\vec{\upsilon}(t+1) = \Upsilon(\vec{\upsilon}(t)) \tag{1}$$

gdzie odwzorowanie Υ jest określone działaniem naszego generatora. Możemy z dobrym przybliżeniem przyjąć, że zachodzi:

$$\Upsilon(\vec{\upsilon}) = L^r \vec{\upsilon} \mod 1,$$

gdzie liniowa transformacja L jest zdefiniowana jako:

$$L\vec{\upsilon} = (\upsilon_1, \upsilon_2, \ldots, \upsilon_{r-1}, \upsilon_{r-s} - \upsilon_0)^T$$

i dlatego też kształt macierzy L jest dosyć prosty:

$$L = \begin{pmatrix} 0 & 1 & 0 & \ldots \\ 0 & 0 & 1 & 0 \\ \vdots & \vdots & \ddots & \\ -1 & \ldots & 1_{(r, r-s)} & \ldots \end{pmatrix}_{(r, r)} \tag{2}$$

Wielkością charakterystyczną dla układów dynamicznych jest tzw. wykładnik Lapunowa. Pokazuje on, czy położone blisko siebie w chwili początkowej trajektorie po upływie pewnego czasu pozostaną bliskie sobie, czy też zaczną się od siebie oddalać. Rozpatrzmy ogólną postać odwzorowania:

$$\upsilon_{n+1} = f(\upsilon_n)$$

i zobaczmy, jak bardzo oddalają się od siebie dwie N-te iteraty $f^N(\upsilon_0)$ oraz $f^N(\upsilon_0 + \epsilon_0)$, dla których punkt startowy różnił się o ϵ_0. Względny wzrost całkowitego błędu ϵ_N możemy zapisać w postaci:

$$\left|\frac{\epsilon_N}{\epsilon_0}\right| = \left|\frac{\epsilon_N}{\epsilon_{N-1}}\right| \cdots \left|\frac{\epsilon_1}{\epsilon_0}\right|.$$

Po zlogarytmowaniu obydwu stron powyższej tożsamości otrzymujemy wykładnik Lapunowa dla odwzorowań dyskretnych:

$$\frac{1}{N} \ln\left|\frac{\epsilon_N}{\epsilon_0}\right| = \frac{1}{N} \sum_{m=1}^{N} \ln\left|\frac{\epsilon_m}{\epsilon_{m-1}}\right|,$$

a jego przybliżone wartości możemy już później szacować, dokonując różnych dodatkowych, upraszczających założeń, dotyczących wielkości ϵ_m.

Dla odwzorowań ciągłych w sposób naturalny przyjmujemy, że:

$$|f^N(\upsilon_0 + \epsilon) - f^N(\upsilon_0)| \simeq \epsilon \exp(N\lambda(\upsilon_0)).$$

Mamy zatem:

$$\lambda(\upsilon_0) = \lim_{N\to\infty} \lim_{\epsilon\to 0} \frac{1}{N} \ln \left|\frac{f^N(\upsilon_0 + \epsilon) - f^N(\upsilon_0)}{\epsilon}\right| = \lim_{N\to\infty} \ln \left|\frac{df^N(\upsilon_0)}{d\upsilon_0}\right|,$$

a w konsekwencji wzór na wykładnik Lapunowa:

$$\lambda(x_0) = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \ln |f'(x_i)|,$$

jeśli tylko skorzystamy z przepisu na różniczkowanie funkcji złożonej. Dodatnia wartość tego wykładnika sygnalizuje rozchodzenie się iterat w przestrzeni stanów – pojawia się *chaos*. Z punktu widzenia własności generatorów liczb losowych jest to sytuacja bardzo pożądana.

Wróćmy zatem do naszego wyjściowego odwzorowania i do odpowiadającej mu macierzy $L_{(r,\ r)}$. Dosyć łatwo wyprowadzić równanie charakterystyczne dla tej macierzy; ma ono postać:

$$\lambda^r - \lambda^{r-s} + 1 = 0$$

i można je bez większych przeszkód rozwiązać w sposób numeryczny. Dla $r = 24$ i $s = 10$ otrzymujemy cztery wartości własne z maksymalną bezwzględną wartością równą[3]

$$|\lambda|_{max} \simeq 1.04299.$$

Jeśli więc śledzimy rozchodzenie się dwóch bliskich sobie w chwili $t = 0$ trajektorii, to w kierunkach odpowiadających tym wartościom własnym dla dużych

[3] Dla takich wartości przesunięć spodziewamy się, że okres generatora będzie rzędu $\simeq 10^{171}$. Trudno tu o systematyczne powtórki: w ciągu miliona lat i tak nie wyprodukujemy za pomocą komputera więcej niż około 10^{40} liczb losowych...

czasów t moduł odległości między tymi trajektoriami będzie się zachowywał w przybliżeniu jak $\propto e^{\mu t}$, gdzie:

$$\mu = r \ln(|\lambda|_{max}) \simeq 1.010.$$

Znalezienie generatorów o większej wartości μ będzie wskazywało na ich wyższą jakość: ze wzrostem liczby iteracji przestrzeń stanów będzie coraz lepiej pokryta. Pomysł na otrzymywanie dobrych ciągów liczb losowych narzuca się teraz nieomal sam: z wygenerowanych podciągów należy zachować do dalszego, właściwego wykorzystania tylko te, których wykładnik Lapunowa jest dodatni[4]. Ten pomysł zrealizowano w generatorach typu RAN(dom)LUX (ury), w których określa się pięć stopni „luksusu", scharakteryzowanych przez pewną liczbę naturalną p, zawartą w przedziale $24 \leq p \leq 389$. Ta ostatnia wartość (jest to liczba pierwsza) charakteryzuje sytuację, kiedy wszystkie 24 bity mantysy mają chaotyczny charakter. Im większa wartość p, tym większy stopień *luksusu*: zaakceptowane liczby losowe przechodzą wtedy coraz surowsze testy statystyczne. Procedura generowania jest następująca:

- generujemy ciąg 24 liczb losowych,
- odrzucamy kolejne $p - 24$ liczby,
- akceptujemy następne 24 liczby itp.

Tak wygenerowane liczby nie będą, mimo wszystko, wolne od *dalekozasięgowych* korelacji; korelacje te będą jednak o 100 rzędów wielkości za małe, by je wykryć... Za ich wysoką jakość trzeba będzie jednak zapłacić pewną cenę: jest nią spowolnienie procesu generowania ciągu liczb losowych. Mimo wszystko nie musi to być cena zabójcza przy obecnych postępach techniki obliczeniowej.

Pierwotnie realizacje komputerowe generatora RANLUX były prezentowane w Fortranie, w tej chwili można je znaleźć także w C/C++[5].

Skoro zostało już wprowadzone do problemu testowania generatorów liczb losowych pojęcie chaotyczności, to warto zwrócić uwagę na jeszcze jeden test generatorów tych liczb związanych z pojęciem dyskretnych układów dynamicznych [11].

Weźmy wektor $\vec{x} = (x_1(t), x_2(t), \ldots, x_n(t))$ opisujący jakiś układ dynamiczny oraz dosyć dowolną (i prostą) funkcję $\Phi(\vec{x}(s))$, np. $\Phi(\vec{x}(t)) = x_1(t)$. Za pomocą tej funkcji konstruujemy dwie inne funkcje pomocnicze:

$$\Theta = ct + \int_0^t ds \Phi(\vec{x}(s)), \quad c > 0$$

[4] Nie zapomnijmy o jeszcze jednym pojęciu przydatnym przy analizie statystycznych własności liczb losowych. Jest to entropia Kołmogorowa = $\sum_i ln(|\lambda_i|)$.

[5] Jedna z takich realizacji pod nazwą ranlxd.c z 2005 roku pochodzi także od M. Lüschera. Jest ona ogólnie dostępna jako GNU General Public Licence i liczy ponad 600 wierszy kodu.

$$p(t) = \int_0^t ds\Phi(\vec{x}(s))cos(\Theta(s))$$

oraz odpowiednią funkcję korelacji:

$$M(t) = \lim_{T\to\infty} \frac{1}{T}\int_0^T d\tau\,(p(t+\tau) - p(\tau))^2, \qquad T \gg t.$$

Okazuje się, że w granicy bardzo dużych czasów dla funkcji:

$$K = \lim_{t\to\infty} \frac{log(M(t)+1)}{log(t)}$$

zachodzi:

$$K \simeq \begin{cases} 1, & chaos \\ 0, & brak\ chaosu. \end{cases}$$

Przedstawiony powyżej zapis dotyczy układów ciągłych (można tu badać np. zachowanie się rozwiązań oscylatora van der Pola), dopasowanie go do układów dyskretnych nie stwarza jednak większych problemów. A jak już wspomnieliśmy wcześniej, na działanie generatorów liczb losowych można spojrzeć właśnie jak na zachowanie się dyskretnych układów dynamicznych.

3. Procesy równoległe – testy

W ostatniej dekadzie pojawił się inny sposób wykrywania (jakby nie było, delikatnych) korelacji w różnych ciągach liczb losowych $\{x_n^{(1)}\}$, $\{x_n^{(2)}\}$, ... $\{x_n^{(r)}\}$, a nie tylko w jednym takim ciągu $\{x_n\}$. Jest to związane z tzw. obliczeniami równoległymi.

Pierwszy z testów nawiązuje do najbardziej chyba znanego problemu błądzenia losowego na siatce dwuwymiarowej[6]. Trudno się więc dziwić, że idea błądzenia przypadkowego została także wykorzystana do testowania generatorów przy obliczeniach *współbieżnych*.

Wyobraźmy sobie wychodzących z jednego węzła dwuwymiarowej sieci dwóch „Jasiów Wędrowniczków", z których każdy jest zaopatrzony w swój własny generator. Generator ten wskazuje mu jeden z 4 kierunków (E, W, N, S), w którym powinien się on poruszać w kolejnych węzłach tej sieci. Możemy so-

[6] W najprostszej wersji powszechnie znanego jako problem „pijanego marynarza" lub też „Jasia Wędrowniczka".

bie zadać następujące pytanie: jakie jest prawdopodobieństwo P_J, że po J krokach drogi tych dwóch „wędrowniczków" się nie przecinają (z wyjątkiem punktu startowego)? Do przecięcia trajektorii nie musi dojść w tym samym czasie. Dla procesu losowego P_J zachowuje się asymptotycznie jak:

$$P_J \simeq J^{-\alpha},\ \alpha = 5/8.$$

Odchylenie α od tej wartości jest pewną miarą jakości użytych generatorów.

Drugi z testów możemy nazwać testem *wysokościowym*. Rozważmy pozycję x_i jednego „skoczka" jako funkcję wykonanej przez niego liczby skoków j. Położenie $x_j = \Sigma_{i=1}^{j} \delta x_i$ jest sumą przesunięć δx_i będących liczbami losowymi określonymi następującym wzorem:

$$\delta x_i = \begin{cases} +1, & o\ ile\ r_i \le \frac{1}{3} \\ 0, & o\ ile\ \frac{1}{3} \le r_i \le \frac{2}{3} \\ -1, & o\ ile\ r_i \ge \frac{2}{3} \end{cases}.$$

Budujemy teraz *dwie* sekwencje liczb losowych $x_i^{(1)}$ oraz $x_i^{(2)}$, korzystając z dwóch niezależnych ciągów liczb losowych $r_i^{(1)}$ oraz $r_i^{(2)}$ o rozkładzie równomiernym, należących do przedziału [0, 1]. Następnie definiujemy różnicę wysokości dla tych dwóch procesów $w_j = x_j^{(1)} - x_j^{(2)}$ i odpowiednią funkcję korelacji $W_j = \langle |w_j - w_0| \rangle \simeq j^{-\alpha}$. Dla dobrych generatorów liczb losowych spodziewamy się $\alpha \simeq \frac{1}{2}$ dla dużych wartości j.

Przytoczone testy nie są, rzecz jasna, jedynymi testami wychodzącymi poza proste obliczanie współczynnika korelacji dla ciągów liczb losowych (por. [12]).

Czytelnik zainteresowany szerzej tematyką generatorów liczb losowych znajdzie wiele interesującego materiału w książeczce [13].

4. Zakończenie

Sztuka budowy dobrych komputerowych generatorów liczb losowych posunęła się naprzód w ostatnich dwóch dekadach dzięki pewnym ideom zaczerpniętym z teorii układów dynamicznych. Trudno przewidzieć, czy w najbliższym czasie nie pojawią się nowe pomysły zmieniające nasz pogląd na budowę dobrych generatorów liczb losowych. Inspiracje, jak widać, mogą być bardzo nieoczekiwane. Problem nie polega na tym, by w nieskończoność testować statystyczne własności nowych generatorów, lecz by budować je, opierając się na jakichś sensownych przesłankach.

Za częściowe wsparcie badań związanych z niniejszą pracą podziękowania należą się *Marie Curie Actions Transfer of Knowledge, project COCOS (contractMTKD-CT-2004517186)*. „Eureka".

Bibliografia

[1] Metropolis N., Ulam S., *The Monte Carlo Method*, „Journal of American Statistical Association" 1949, *44*, 335.

[2] Glasserman P., *Monte Carlo Methods in Financial Engineering*, New York 2004.

[3] Knuth D.E., *Sztuka programowania. Algorytmy seminumeryczne*, t. 2, Warszawa 2002.

[4] Hamilton K.G., James F., *Acceleration of RANLUX*, „Computer Physics Communications" 1997, *101*, 241.

[5] Hamilton K.G., *Assembler RANLUX for PCs*, „Computer Physics Communications" 1997, *101*, 249.

[6] Lehmer D.H., *Mathematical Methods in Large-scale Computing Units*, p. 141 [w:] *Proceedings of the Second Symposium on Large Scale Digital Computing Machinery*, Cambridge, MA, 1951.

[7] Forsythe G.E., Malcolm M.A., Moler C.C., *Computer Methods for Mathematical Computations*, Englewood Clifs, NJ, 1977.

[8] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., *Numerical Recipes in C*, Cambridge, MA, 1992.

[9] Marsaglia G., Zaman A., *A New Class of Random Number Generators*, „The Annals of Applied Probability" 1991, *1*, 462.

[10] LüscherM., *A Portable High-quality Random Number Generator for Lattice Field Theory Calculations*, „Computer Physics Communications" 1994, *79*, 100.

[11] Gottwald G.A., Melbourne I., *A New Test for Chaos in Deterministic Systems*, „Proceedings of the Royal Society", London 2004, *A 460*, 603.

[12] Vattulainen I., *Framework for Testing Random Numbers in Parallel Calculations*, „Physical Review" 1999, *59*, 7200.

[13] Wieczorkowski R., Zieliński R., *Komputerowe generatory liczb losowych*, Warszawa 1997.

[14] James F., *RANLUX: A Fortran Implementation of the High-quality Pseudo-random Number Generator of Lüscher*, „Computer Physics Communications" 1994, *79*, 111.

[15] Gottwald G.A., Melbourne I., *Testing for Chaos in Deterministic Systems with Noise*, „Physica A" 2005, *21*, 100.

Andrzej Łachwa

Podobieństwo zbiorów

1. Wstęp

Niemal codziennie używamy określenia „podobieństwo" i wskazujemy rzeczy podobne do siebie. Na pierwszy rzut oka jest to pojęcie proste. Jednak informatyk patrzy na takie z pozoru proste pojęcia nieco inaczej. Musi nadać im tak precyzyjny sens, by nadawały się na elementy budowanego modelu rzeczywistości lub na operacje wchodzące w skład tworzonej metody obliczeniowej.

Pojęcie podobieństwa wykorzystujemy głównie do budowania zbiorów: zbiór składa się z elementów podobnych! I choć określenie „zbiór" zastępujemy często terminami „typ encji" czy „klasa obiektów", nie zmienia to istoty sprawy. Łącząc elementy w zbiór, podejmujemy decyzję, na czym ma polegać ich podobieństwo.

Jak już to wyjaśniałem w poprzednim tomie „Informatyki" (zob. [4]), zbiór może być pojmowany na wiele sposobów. Tutaj zajmę się pojęciem podobieństwa, w szczególności zaś podobieństwa zbiorów.

2. Podobieństwo obiektów

Równość w matematyce jest eksplikowana przez relację równoważności. Relacja ta to relacja jednocześnie zwrotna, symetryczna i przechodnia. Dzieli ona uniwersum, na którym jest określona, na klasy równoważności: klasy obiektów równoważnych. Podział taki jest rozłączny i zupełny.

Równość (jednakowość, identyczność) jest zatem przede wszystkim binarną relacją równoważności określoną na pewnym uniwersum. Ponadto jest wartością względną, zależy od sytuacji czy punktu widzenia obserwatora (na danym uniwersum można zwykle zdefiniować różne relacje równoważności). I wreszcie równość oznacza zastępowalność jednego obiektu drugim w określonej sytuacji (por. [5, s. 43–49]).

Podobieństwo oznacza tylko częściową zastępowalność – istnieje możliwość zastąpienia jednego obiektu drugim, ale z pewnym ryzykiem czy pewną stratą. Podobieństwo obiektów danego uniwersum w matematyce nazywa się tolerancją i jest ono relacją zwrotną i zarazem symetryczną. Przechodniość nie jest wymagana, a to dlatego, że obiekty podobne nie są identyczne, nieznacznie różnią się od siebie i te drobne różnice między kolejnymi podobnymi obiektami mogą doprowadzić do obiektów całkowicie różnych od tych początkowych (por. [5, s. 69 i n.]).

Przykładem tego jest znana zabawa ze słowami polegająca na przekształceniu słowa początkowego w słowo końcowe przez zmianę w każdym kolejnym słowie znajdującym się pomiędzy nimi tylko jednej litery, np. możemy w taki sposób przekształcić słowo „kot" w słowo „lew" ([5, s. 72]): *kot – kos – los – lis – lin – len – lew*.

Zbiór U z określoną na nim relacją tolerancji T nazywa się przestrzenią tolerancji. Struktura tej przestrzeni jest bardzo ciekawa. W szczególności okazuje się, że dowolną tolerancję można określić za pomocą zbioru cech elementów uniwersum U w taki sposób, że elementami podobnymi są te, które mają co najmniej jedną wspólną cechę (por. [5, s. 79 i n.]).

3. Podobieństwo zbiorów dystrybutywnych

Dwa zbiory dystrybutywne są równe, gdy mają te same elementy. Zbiory te są równoważne, gdy można wskazać pewne identyczne cechy tych zbiorów, np. równoliczność. Jeżeli osłabimy to wymaganie przez rezygnację z przechodniości, to otrzymamy relację tolerancji. A zatem, tak jak wyżej, jeżeli mamy dany zbiór cech, to możemy przyjąć, że zbiory są podobne, gdy mają co najmniej jedną wspólną cechę.

W matematyce podobieństwo zbiorów dystrybutywnych często definiuje się odmiennie, jako szczególną relację równoważności – *równokształtność*. Na przykład w geometrii dwa wielokąty uznaje się za podobne, gdy mają te same kąty i proporcje: mają taki sam kształt, ale mogą mieć różną wielkość. W algebrze dwa wyrażenia nazywa się podobnymi, gdy mają ten sam kształt z dokładnością do współczynników liczbowych. Przykłady takie można mnożyć.

4. Podobieństwo zbiorów rozmytych

Zbiorem rozmytym jest ogół tych elementów pewnego uniwersum, które można powiązać myślowo w całość na podstawie jakieś ich własności, zwykle nieostrej (por. [3, s. 12]).

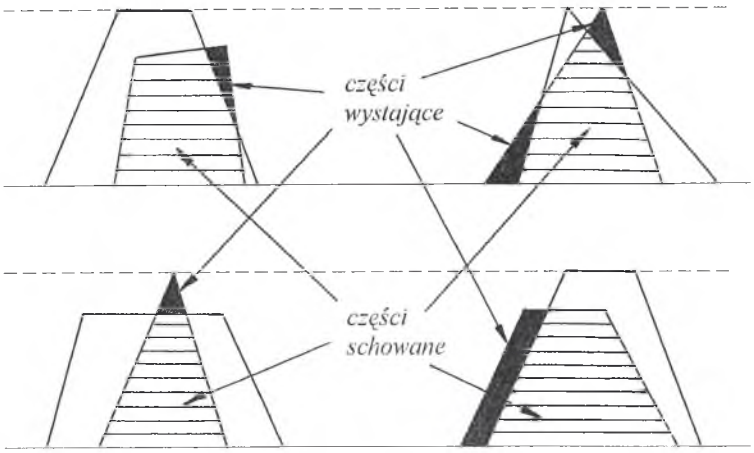
Dwa zbiory rozmyte określone na pewnym uniwersum są identyczne wtedy i tylko wtedy, gdy każdy element tego uniwersum należy do tych zbiorów w tym samym stopniu. Podobnie definiuje się inkluzję zbiorów rozmytych: zbiór rozmyty A jest zawarty w zbiorze rozmytym B (określonym na tym samym uniwersum, co A) wtedy i tylko wtedy, gdy każdy element uniwersum należy do zbioru A w stopniu nie większym, niż należy do zbioru B (por. [6]).

Tak rozumiane równość i inkluzja zbiorów rozmytych są wprawdzie formalnie poprawne i eleganckie, ale mają niewielkie znaczenie z punktu widzenia metod obliczeniowych. Po pierwsze, dlatego że w przypadku uniwersum nieskończonego (lub skończonego, ale bardzo dużego – por. [4, s. 37]) nie moglibyśmy sprawdzić, czy odpowiednia relacja zachodzi; po drugie zaś, dlatego że zdefiniowane wyżej dwie ostre własności zwykle nie mają zastosowania przy przetwarzaniu rozmytej informacji. Rozmytość to przecież niewyraźność i niepewność. Niewielkie różnice w sposobie rozumienia tej niepewności (niewielkie różnice w wartościach funkcji przynależności) nie powinny decydować o tak istotnych własnościach, jak równość i zawieranie. Zaproponowano zatem (por. [1, s. 43–46]) rozmycie własności identyczności i inkluzji zbiorów rozmytych. Jednak moim zdaniem, zamiast mówić o rozmytej równości zbiorów (identyczności zbiorów), lepiej stosować termin „podobieństwo". Podobieństwo rozumiane jest bowiem w języku naturalnym jako własność nieostra i niekoniecznie przechodnia, równość zaś (czy też identyczność) – jako własność ostra i przechodnia.

4.1. Podobieństwo oparte na rozmytej inkluzji

Rozmycie ostrego znaczenia inkluzji sprowadza się do tego, że własność zawierania się jednego zbioru w drugim będzie rozumiana jako stopniowalna: od stopnia 0 oznaczającego niezawieranie, poprzez przypadki zawierania częściowego, do stopnia 1 oznaczającego zawieranie całkowite. Dodatkowo oczekuje się, że taka rozmyta własność będzie zwrotna ($A \subset A$ w stopniu 1) i antysymetryczna (jeżeli $A \subset B$ w stopniu α i $B \subset A$ w stopniu β, to A jest podobne do B w stopniu $\alpha \wedge \beta$).

Oryginalny sposób obliczania rozmytej inkluzji wprowadziłem w pracy [3]. Zasadzał się on na spostrzeżeniu, że stopień częściowego zawierania się zbioru A w zbiorze B ma nas informować o tym, jak mają się do siebie części zbioru A, które „wystają" poza zbiór B, do części zbioru A, które „mieszczą się wewnątrz" zbioru B. Wzrokowo potrafimy to łatwo ocenić (por. rysunek 1) i wydaje się, że porównujemy przede wszystkim maksymalne odległości odpowiednich krzywych i powierzchnie odpowiednich figur (na rysunku 1 powierzchnie te zaczerniono i zakreskowano).



Rysunek 1. Części wystające i części schowane

Założyłem, że wzór na obliczanie stopnia zawierania się zbiorów powinien być na tyle prosty, by dało się go łatwo stosować w rozmaitych sytuacjach praktycznych. Nie możemy więc obliczać powierzchni, długości krzywych czy maksymalnych odległości między dowolnymi krzywymi.

Zaproponowany wzór dotyczy zbiorów rozmytych o funkcjach przynależności ciągłych i rzeczywistych, a ponadto odnosi się tylko do zbiorów, których nośniki są ograniczone. Niektóre z tych założeń można łatwo ominąć.

W omawianym wzorze wykorzystuję dwa łatwe do policzenia wskaźniki – wysokość oraz nośnik. Przypomnijmy:

- wysokością zbioru rozmytego A nazywa się najwyższy stopień przynależności i oznacza się ją przez $h(A)$,
- nośnikiem zbioru rozmytego A nazywa się podzbiór tych elementów uniwersum, dla których stopień przynależności jest niezerowy, i oznacza przez $supp(A)$.

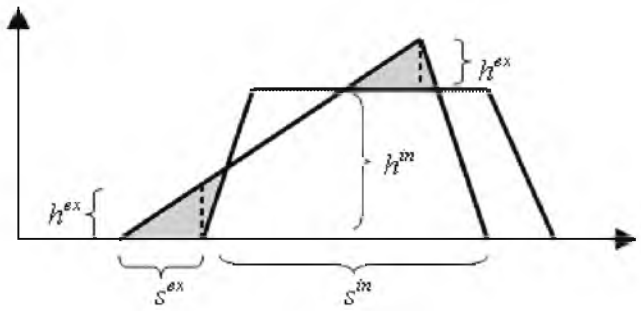
W omawianym wzorze wziąłem pod uwagę wysokość h^{ex} różnicy ograniczonej $A - B$, wysokość h^{in} iloczynu $A \cap B$, długość s^{in} nośnika części schowanej i długość s^{ex} nośnika części wystającej (por. rysunek 2).

Z pełnym zawieraniem A w B będziemy mieć do czynienia wtedy, gdy $h^{ex} = 0$ i $s^{ex} = 0$, czyli gdy A nigdzie nie wystaje poza B. Z brakiem zawierania mamy do czynienia w dwóch przypadkach: (1) wtedy gdy $h^{in} = 0$; (2) wtedy gdy $h^{ex} \geq h^{in}$ lub $s^{ex} \geq s^{in}$, przy czym ten drugi przypadek jest propozycją arbitralną i rozumiem, że może być ona dyskusyjna. Jeżeli jednak zgodzimy się z tym rozstrzygnięciem, to z częściowym zawieraniem będziemy mieć do czynienia w pozos-

tałych przypadkach, tzn. wtedy gdy $0 \leq s^{ex} < s^{in}$ i zarazem $0 < h^{ex} < h^{in}$. Stopień taki może być liczony za pomocą każdego wzoru typu:

$$(1 - s^{ex} / s^{in})\ T\ (1 - h^{ex} / h^{in}),$$

gdzie T jest dowolną trójkątną operacją mnożenia (por. [3, s. 47–50), np. operacją minimum.



Rysunek 2. Wysokości i nośniki części schowanych i części wystających

Ostatecznie wzór na obliczanie stopnia zawierania się zbioru rozmytego A w zbiorze rozmytym B ma postać:

$$dg(A \subset B) = \begin{cases} 1 & \text{dla } h^{ex} = 0 \\ 0 & \text{dla } h^{ex} \geq h^{in} \text{ lub } s^{ex} \geq s^{in} \text{ lub } h^{in} = 0. \\ (1 - s^{ex} / s^{in})\ T\ (1 - h^{ex} / h^{in}) & \text{wpp} \end{cases}$$

Równość dwóch zbiorów ostrych zachodzi wtedy, gdy pierwszy z nich zawiera się w drugim, a drugi w pierwszym. Podobieństwo zbiorów rozmytych mogę więc zdefiniować jako mniejszy ze stopni tych dwóch rozmytych inkluzji:

$$dg(A \approx B) = dg(A \subset B) \wedge dg(B \subset A).$$

4.2. Podobieństwo oparte na metryce

Metryką na przestrzeni zbiorów rozmytych $F(X)$ na uniwersum X nazywamy funkcję m: $F(X) \times F(X) \to R^{+} \cup \{0\}$, taką, że:

- $m(A, B) \geq 0$
- $m(A, B) = m(B, A)$
- $A = B \Rightarrow m(A, B) = 0$
- $m(A, C) \leq m(A, B) + m(B, C)$.

Dla zbiorów rozmytych na uniwersum skończonym zaproponowano różne rodzaje odległości (por. [1, 2, 3]). Nas będą interesować tylko metryki znormali-

zowane, czyli przyjmujące wartości w przedziale [0, 1]. Przykładem takiej metryki jest tzw. odległość dwudzielna:

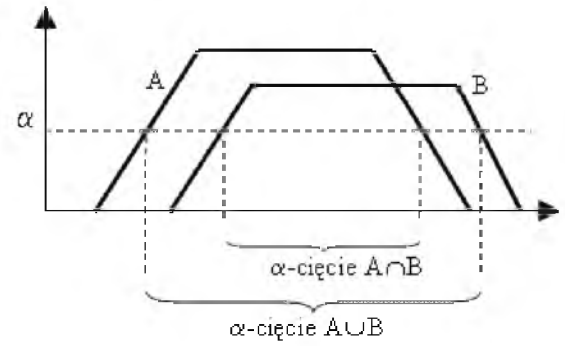
$$k(A, B) = 1 - |A \cap B| / |A \cup B|.$$

Wyznaczanie podobieństwa dwóch zbiorów rozmytych może polegać na obliczaniu dopełnienia ich znormalizowanej odległości. Stopień podobieństwa zbiorów rozmytych A i B to wówczas $E(A, B) = 1 - m(A, B)$, gdzie m jest dowolną metryką znormalizowaną. Na przykład dla metryki dwudzielnej mamy:

$$E(A, B) = |A \cap B| / |A \cup B|.$$

4.3. Podobieństwo oparte na przekrojach

Przyjmijmy, że X jest zbiorem liczb rzeczywistych, nośniki zbiorów rozmytych A i B są ograniczone, a każde α-przecięcie iloczynu i sumy tych zbiorów jest odcinkiem, skończoną sumą odcinków, zbiorem jednoelementowym (przecięcie na poziomie wartości szczytowej) albo zbiorem pustym (przecięcie powyżej wysokości). Przy takich założeniach możemy we wzorze $E(A, B) = |A \cap B| / |A \cup B|$ zastąpić iloczyn i sumę zbiorów rozmytych ich α-przekrojami, moce zaś tych zbiorów – ich długościami (długością zbioru ostrego będącego sumą odcinków jest suma długości tych odcinków). Stosunek długości α-cięcia iloczynu do długości α-cięcia sumy pokaże nam na każdym poziomie podobieństwo między zbiorami A i B (por. rysunek 3).

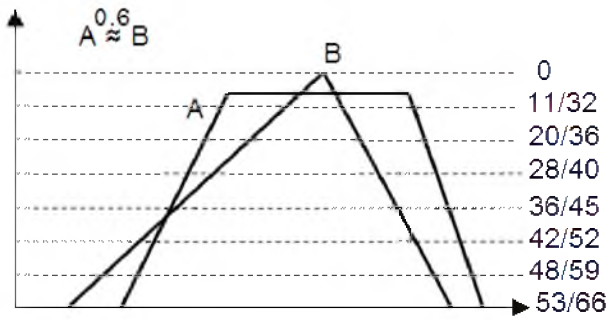


Rysunek 3. Przekrój sumy i iloczynu dwóch zbiorów na poziomie α

Jeśli policzymy średnią z tych liczb, to dla zbiorów identycznych uzyskamy wartość 1, dla zbiorów o pustym iloczynie – wartość 0, dla pozostałych zaś przypadków – stopnie dobrze oddające intuicyjnie pojmowane podobieństwo (por. rysunek 4).

W zastosowaniach praktycznych możemy ograniczyć się do przecięcia sumy i iloczynu zbiorów rozmytych A i B na skończonej, niewielkiej liczbie n poziomów:

$$E^n(A, B) = (1/n) \cdot \sum_{\alpha = 0,\ \lambda,\ 2\lambda,\ \ldots,\ n\lambda} |(A \cap B)_\alpha| \ / \ |(A \cup B)_\alpha|, \text{ gdzie } \lambda = [h(A) \vee h(B)] \ / \ n.$$



Rysunek 4. Stopień podobieństwa zbiorów dla $n = 8$

Podobieństwo zbiorów A i B z rysunku 4 policzono na dwa sposoby: dla $n = 8$ i dla $n = 80$. Otrzymane wyniki wynoszą 0,601 i 0,602, czyli są dostatecznie podobne, aby nie stosować tak wielu przecięć, jak w drugim z tych sposobów. W przypadku zbiorów łamanych (zbiorów rozmytych, których kształt przybliżono łamaną, por. [4, s. 45]) poziomy przecięć raczej nie powinny być rozłożone symetrycznie, tylko przechodzić przez wierzchołki łamanych.

5. Podsumowanie

Podobieństwo zbiorów dystrybutywnych jest często definiowane w matematyce klasycznej (np. w geometrii czy algebrze) niezbyt szczęśliwie, jako relacja równoważności. W matematyce zbiorów rozmytych eksplikacja pojęcia podobieństwa jest w pełni zgodna z jego znaczeniem w języku naturalnym i odpowiada relacji tolerancji.

Bibliografia

[1] Kacprzyk J., *Zbiory rozmyte w analizie systemowej*, Warszawa 1986.
[2] Lin C.-T., Lee C.S.G., *Neural Fuzzy Systems. A Neuro-Fuzzy Synergism to Intelligent Systems*, New York 1996.
[3] Łachwa A., *Rozmyty świat zbiorów, liczb, relacji, faktów, reguł i decyzji*, Warszawa 2001.

[4] Łachwa A., *Który zbiór wybrać?*, „Acta Academiae Modrevianae. Informatyka", Kraków 2006, s. 35–49.
[5] Szrejder J.A., *Równość, podobieństwo, porządek*, Warszawa 1975.
[6] Zadeh L.A., *Fuzzy sets*, „Information and Control" 1965, vol. 8, s. 338–353.

Marek Szepski

Modelowanie zarządzania danymi w bazach danych

1. Problem modelowania zarządzania danymi

Modelowanie zarządzania danymi rzadko pojawia się w literaturze poświęconej bazom danych. Jest jednak oczywiste, że dane w nich przechowywane są zarządzane i dzieje się to zgodnie z jakimś modelem. W relacyjnym systemie zarządzania bazą danych (RSZBD) model wynika z reguł sformułowanych przez Edgara Codda. Przyjmowany jest on wtedy jako „naturalny, wbudowany w bazę danych" i nie zawsze uświadamiamy sobie, jak jest sztywny. Niewątpliwe zalety RSZBD okupione są niestety wieloma ograniczeniami.

Dominujące obecnie modelowanie obiektowe koncentruje się na analizie dziedziny problemu. Analiza diagramów klas typowych przykładów prezentowanych w literaturze przedmiotu pokazuje, że zarządzanie danymi jest tam pomijane. Autorzy koncentrują się na modelowanym problemie, abstrahując od zagadnień związanych z tworzeniem systemu. Można przypuszczać, że w wielu przypadkach system taki byłby konstruowany jako baza hybrydowa obiektowo-relacyjna i zarządzanie danymi odbywałoby się według schematu relacyjnego. Modelowanie obiektowe daje jednak większe możliwości niż tylko proste sprowadzenie do znanych schematów RSZBD.

Baz danych budowanych na podstawie modelu obiektowego nie ograniczają sztywne standardy w rodzaju reguł Codda. Daje to twórcom systemu możliwość dowolnego modelowania rozwiązywanego problemu, ale zobowiązuje także do określenia modelu zarządzania danymi.

2. Modelowanie zarządzania danymi w RSZBD

W przypadku budowania bazy danych na podstawie RSZBD problem modelowania zarządzania danymi został rozwiązany „odgórnie". Reguły sformułowa-

ne przez Edgara Codda precyzyjnie określają wszystkie aspekty RSZBD, a niektóre szczegółowo odnoszą się do zarządzania danymi [1, 6]. Przyjęto, że mianem RSZBD określa się jedynie te systemy, które są oparte na regułach Codda. Przeanalizujmy konsekwencje wybranych reguł.

> Reguła 1: Wszystkie informacje znajdujące się w relacyjnej bazie danych są reprezentowane wyłącznie na poziomie logicznym i w jednolity sposób – jako wartości w tabelach.

Ta prosta reguła ma daleko idące konsekwencje. Po pierwsze: nie mamy bezpośredniego dostępu do danych zapisanych na dysku. Decyzje o zapisie danych, sposobie zapisu czy momencie modyfikacji są przekazane systemowi zarządzania bazą danych i realizowane automatycznie. Takie podejście zapewnia bezpieczeństwo danych, zdejmuje z projektantów i użytkowników obowiązek pamiętania o utrwaleniu danych oraz uniemożliwia ich zmianę z pominięciem systemu zarządzania. Inaczej mówiąc, w bazie danych można wyróżnić dwie warstwy: warstwę tabel (struktur logicznych) dostępną dla użytkownika i warstwę danych (fizyczną, trwale zapisaną). Komunikację miedzy tymi warstwami zapewnia system zarządzania, na który użytkownik nie ma wpływu. Projektując bazę danych, nie musimy, a nawet więcej, nie możemy modelować zarządzania danymi, gdyż robi to za nas RSZBD.

Po drugie: reguła ta ogranicza struktury reprezentujące dane wyłącznie do tabel. Z punktu widzenia struktur danych możemy patrzeć na tabelę jako na wektor rekordów. W połączeniu z regułą 2 stanowi to duże ograniczenie.

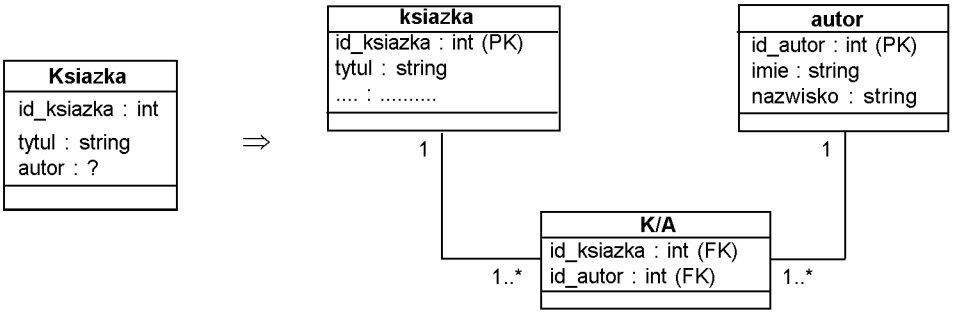
> Reguła 2: Każda jednostka informacji (wartość atomowa, niepodzielna) w RSZBD musi być dostępna w sposób logiczny, przez odwołanie realizowane za pomocą kombinacji: nazwy tablicy, wartości klucza głównego i nazwy kolumny.

Ta reguła dotyczy przede wszystkim modelowania dziedziny problemu, ale z punktu widzenia zarządzania danymi, ze względu na wymóg ich niepodzielności, ogranicza je praktycznie do typów prostych. Oczywiście dostawcy RSZBD wprowadzają wiele różnych typów zmiennych, np.: rozmaite typy liczbowe, różnorodne łańcuchy, różne postacie dat (pamiętajmy, że data jest liczbą, a typ oznacza jedynie sposób wyświetlania) czy nawet tablice, adresy URL, pliki itd. Zasada niepodzielności oznacza jednak, że dostęp do części składowych informacji wymaga napisania dodatkowych funkcji.

Konsekwencją tych, a także innych, nieprzywołanych tu reguł jest wymaganie określone w regule 5, którą można by streścić następująco: istnieje język taki jak SQL i jest on wymaganą częścią RSZBD. W systemie mamy więc narzędzia

służące do definiowania i przetwarzania danych, kontrolujące dostęp użytkowników, przebieg transakcji i integralność danych.

Integralność danych jest kluczowa dla użytkowników bazy danych. Reguły Codda 10 i 12 mówią wprost, że więzy integralności są częścią RSZBD (a nie aplikacji użytkownika) i są definiowane wraz z definiowaniem danych oraz że nie jest dopuszczalne takie przetwarzanie danych, które mogłoby usunąć lub obejść te więzy. Konsekwencją chęci zachowania integralności danych jest proces normalizacji tabel, który wymaga ich podziału na nowe tabele i prowadzi do zwiększenia ich liczby oraz zmniejszenia czytelności modelu. Rezultatem normalizacji jest powstanie tabel, które nie mają odpowiedników w istniejących obiektach. Rysunek 1 pokazuje typowe efekty poradzenia sobie ze związkiem „wiele do wielu", czyli ze złożonymi wartościami jednego atrybutu. Ponieważ książka może mieć wielu autorów, to musimy stworzyć dodatkową tabelę autorów i tabelę łączącą. Mamy więc książki (czy to jest jeszcze książka?) pozbawione autorów, autorów, o których nie wiadomo, czy coś napisali, a rzeczywista książka ukrywa się pod kilkoma (!) rekordami w kompletnie nieczytelnej tabeli łączącej.



Rysunek 1. Normalizacja tabeli „Książka"

Integralność jest budowana na poziomie logicznym, czyli na poziomie modelowania dziedziny problemu, a nie na poziomie zarządzania danymi. Reguły opisujące wymaganie integralności dotyczą w konsekwencji takich zagadnień, jak wybór kluczy głównych i obcych czy przetwarzanie danych o wartościach nieokreślonych (NULL).

Jak widać, wymagania narzucone RSZBD są bardzo duże. Oparcie RSZBD na teorii, zdefiniowanie operacji na danych jako działań algebry relacyjnej i warunków, które muszą być spełnione, aby otrzymywane wyniki tych działań były poprawne, dało bardzo efektywne narzędzie baz danych.

3. Różnice w modelowaniu obiektowym i relacyjnym

Modelowanie obiektowe stało się obecnie standardowym podejściem do modelowania systemów. Nie narzuca ono ograniczeń związanych z modelem relacyjnym, jest bardziej od niego elastyczne i pozwala budować modele powiązane w sposób naturalny z rzeczywistością.

Podstawowe dwa pojęcia modelowania: „encja" w modelu relacyjnym i „obiekt" w modelu obiektowym są podobne. Encja jest na przykład definiowana następująco [1]:

> Encja jest osobą, miejscem, rzeczą lub pojęciem, które posiada cechy interesujące z punktu widzenia organizacji i o którym chce się przechowywać informacje.

Obiekt może być definiowany następująco [7]:

> Obiekt to każdy byt – pojęcie lub rzecz – mający znaczenie w kontekście rozwiązywania problemu w danej dziedzinie przedmiotowej.

Mimo dużego podobieństwa definicji encji i obiektu widać istotną różnicę. W przypadku obiektu nie mówi się o przechowywaniu informacji, które to zadanie jest tylko jednym spośród wielu, jakie stawia się obiektom.

Uogólnieniem pojęcia obiektu jest klasa, a uogólnieniem terminu „encja" są tabele. Celowo użyto tutaj liczby mnogiej (tabele): na rysunku 1 encja „książka" została zamodelowana jako 3 tabele. W innych przypadkach tabela może być zbiorem encji. Porównanie pojęć „tabela" i „klasa" obrazuje różnice między omawianymi podejściami do modelowania.

Tabela 1. Porównanie pojęć „tabela" i „klasa"

Tabela	Klasa
Jest wzorcem oraz zbiorem obiektów.	Jest tylko wzorcem obiektów.
Zawiera trwałe rekordy.	Tworzy ulotne lub trwałe obiekty.
Musi zawierać atrybuty.	Są klasy nieposiadające atrybutów.
Atrybuty są elementarne.	Atrybuty mogą być elementarne lub złożone.
Rekord musi posiadać unikalny identyfikator-klucz.	Obiekty danej klasy są zawsze rozróżnialne, niezależnie od wartości atrybutów.
Jest pasywna.	Jest aktywna.

Jak widać, różnice między tabelą i klasą są ogromne.

Definiując tabelę, określamy strukturę rekordów. Równocześnie powstaje opakowanie zbiorcze rekordów, które pozwala odszukać utworzone rekordy. Niestety, obiekty utworzone przez klasy są jak wolne elektrony w przestrzeni i wymagają stworzenia innych struktur, w których będą przechowywane.

Inaczej też zachowują się rekordy i obiekty w chwili zakończenia działania aplikacji. Obiekty istnieją niestety jedynie w pamięci komputera i koniec pracy programu jest jednocześnie końcem ich istnienia. Koncepcja obiektowych baz danych oparta jest na trwałych obiektach tworzonych przez klasy określone stereotypem <<persistence>>. Narzędzia CASE (np. Visual Paradigme) potrafią klasy <<persistence>> diagramu klas przekształcić automatycznie w diagram ERD. W modelu relacyjnym zadaniem encji jest przechowywanie danych. Zadania powierzane obiektom mogą być bardziej skomplikowane. Istnieją obiekty sterujące, kontrolujące przebieg programu, obiekty graniczne, odpowiedzialne za przekazywanie informacji i oczywiście obiekty przechowujące. Trwałość dotyczy jedynie części modelu, a dokładniej – klas przechowujących.

W modelu obiektowym istnieją klasy abstrakcyjne, tzn. takie, które nie tworzą obiektów. Dotyczy to klas niemających atrybutów (np. interfejs) lub stanowiących tylko wzorzec określający atrybuty lub operacje, wykorzystywany przez klasy dziedziczące. Takie klasy są ważne dla pełnego opisu systemu, nie posiadają jednak danych wymagających przechowywania.

Istotne różnice można znaleźć wśród dopuszczalnych typów atrybutów. Atrybuty obiektów mogą być strukturami złożonymi, a dokładniej – mogą być typu określonej klasy, zarówno zdefiniowanej przez programistę, jak i standardowej. Mogą to być w szczególności:

- zbiór, który zawiera nieuporządkowaną grupę elementów tego samego typu,
- worek (ang. *bag*), który od zbioru odróżnia to, że może zawierać powtarzające się elementy (duplikaty),
- lista – uporządkowana grupa elementów tego samego typu,
- tablica – taka jak w językach programowania, ale o dynamicznym rozmiarze, do której elementów dostęp jest poprzez ich pozycję,
- słownik, składający się z par, tworzonych przez uporządkowany klucz i skojarzoną z nim wartość.

Złożone typy danych są potrzebne między innymi do zapewnienia nawigacji między obiektami. W przykładowej klasie „Książka" atrybut „autor" mógłby być np. listą i informacje o autorze mogłyby być dostępne po odwołaniu się do poszczególnych elementów tej listy.

Innym powodem stosowania złożonych atrybutów jest zaprojektowanie dla obiektów utworzonych przez daną klasę nowej klasy, tzw. klasy kolekcji, przechowującej informacje o wszystkich tych obiektach. Klasa kolekcji pozwala uzyskać dostęp do wszystkich obiektów kolekcji.

Odwołania między obiektami są nawigacyjne. Oznacza to, że przetwarzanie informacji musi być wcześniej zaplanowane i zakodowane w programie. Tworzenie nowych zapytań do bazy w trakcie jej działania jest bardzo utrudnione. Wykorzystanie mechanizmów relacyjnych (np. zapytań SELECT z SQL) w bazie danych wymaga określenia kluczy głównych (patrz: reguła 2 Codda) i przechowywania danych w tabelach. W rezultacie można otrzymać efektywny mechanizm wyszukiwania informacji w bazie (relacyjnej lub hybrydowej).

4. Modele zarządzania danymi w modelowaniu obiektowym

Problem modelowania zarządzania danymi wiąże się z najczęściej obecnie stosowanym podejściem do modelowania systemów, opartym na języku UML (Unified Modeling Language). UML dostarcza bogatego zestawu narzędzi do obrazowania i specyfikowania systemów, obejmującego zarówno dynamikę systemu, jak i jego strukturę statyczną. Oferuje obecnie 13 standardowych diagramów, które modelują różnorodne aspekty analizowanego systemu. Jednocześnie UML jest elastyczny i rozszerzalny, co pozwala dostosować go do specyficznych potrzeb konkretnego problemu. UML powstał z potrzeby prezentowania rozbudowanych systemów w sposób zrozumiały dla szerokiego grona osób współpracujących.

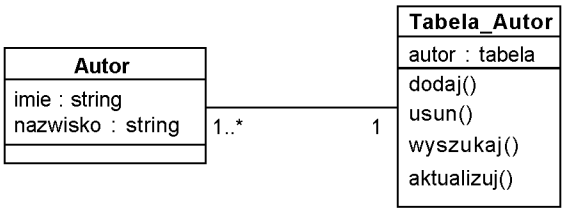
UML jest pomyślany w taki sposób, że koncentruje się na modelowanym systemie i jego złożonych aspektach, nie uwzględniając ograniczeń wprowadzanych przez narzędzia tworzenia aplikacji. Umożliwia to podjęcie decyzji o wyborze narzędzi programowania po zbudowaniu modelu i daje szansę wyboru tych najlepszych. Analiza diagramu klas, uwzględniająca np. dziedziczenie i inne struktury obiektowe, pozwala zdecydować, czy budujemy bazę opartą na narzędziach relacyjnych, obiektowych, czy bazę hybrydową, łączącą oba modele. Nawet jeśli z góry zakładamy, że będziemy tworzyć bazę danych opartą na jakimś RSZBD, to i tak mamy najpierw diagram klas, na podstawie którego budujemy diagram związków encji (ERD). Daje to obraz ograniczeń wprowadzonych przez model relacyjny.

Przekształcenie diagramu klas w diagram związków encji wymaga dokonania kilku operacji. Podobieństwa między tymi diagramami mogą sugerować, że jest to łatwe, ale tak być nie musi. Pierwszym krokiem powinno być zastąpienie złożonych atrybutów dodatkowymi tabelami. Elementem odróżniającym klasy od tabel są operacje. Pokusa ich prostego pominięcia prowadzi do nieuzasadnionego uproszczenia modelu. Część operacji dotycząca zapisywania lub modyfikowania danych może być zamieniona w triggery (procedury zapisane jako element

RSZBD i wyzwalane automatycznie), pozostałe staną się elementem warstwy reguł biznesowych.

Dalsze postępowanie jest już standardem przy modelowaniu relacyjnym i obejmuje: określenie kluczy głównych, dodanie tabel łączących, które zlikwidują związki „wiele do wielu", określenie kluczy obcych i związków, a także ograniczeń oraz przeprowadzenie normalizacji tabel.

Jeżeli chcemy zachować model obiektowy, a RSZBD użyć jedynie do przechowywania i wyszukiwania danych, to prostym rozwiązaniem jest uzupełnienie modelu o klasy realizujące połączenie z RSZBD. Na przykład klasa „Autor" jest powiązana z klasą „Tabela_Autor" (rysunek 2). Operacje tej nowej klasy korzystają z bibliotek odwołujących się do języka SQL, który jest wbudowany w RSZBD, i realizują funkcje zarządzania danymi.



Rysunek 2. Klasa realizująca łączność z relacyjną bazą danych

Inny, bardziej rozbudowany model został zaproponowany w książce C. Argili i E. Yourdona [2]. Model wykorzystuje elementy modelowania obiektowego: dziedziczenie i agregację. Model systemu obejmuje klasy zawierające charakterystykę tabel oraz kolumn. Klasy przechowujące dane dziedziczą zarówno po klasie opisującej obiekty, jak i po abstrakcyjnej klasie charakteryzującej kolumny, a następnie zagregowanej w klasę opisującą tabele, która dodatkowo dziedziczy po klasie opisującej tabele. Celem takiego podejścia jest możliwość wielokrotnego użycia i łatwość modyfikacji, choć odbywa się to kosztem czytelności modelu.

Wybór modelu zarządzania danymi wymaga rozważenia najprostszego, czysto obiektowego rozwiązania. Złożone struktury atrybutów mogą być wprost zapisane w sposób trwały. Przy dużej liczbie danych określenie wprost jest nieprecyzyjne, ale można wówczas skorzystać z któregoś z gotowych systemów obiektowych baz danych, który rozwiąże za nas powstające problemy.

Bibliografia

[1] Allen S., *Modelowanie danych*, Gliwice 2006.
[2] Argila C., Yourdon E., *Analiza obiektowa i projektowanie*, Warszawa 1999.
[3] Booch G., Rumbaugh J., Jacobson I., *UML. Przewodnik użytkownika*, Warszawa 2002.
[4] Graham I., *Metody obiektowe w teorii i w praktyce*, Warszawa 2004.
[5] Harrington J.L., *Obiektowe bazy danych dla każdego*, Warszawa 2000.
[6] Whitehorn M., Marklyn B., *Relacyjne bazy danych*, Gliwice [2003].
[7] Wrycza S., Marcinkowski B., Wyrzykowski K., *Język UML 2.0 w modelowaniu systemów informatycznych*, Gliwice 2005.

Aneta Januszko-Szakiel

Rola migracji i emulacji w strategii długoterminowej archiwizacji publikacji elektronicznych

1. Wstęp

Publikacje elektroniczne stanowią obecnie pokaźną część światowych zasobów bibliotecznych, archiwalnych i muzealnych. Biorąc pod uwagę pogląd, że „technologia cyfrowa ulegnie trwałemu wbudowaniu w dorobek obecnego i następnych pokoleń rodzaju ludzkiego" [1], warto ustalić, co to oznacza dla instytucji pamięci, odpowiedzialnych za zabezpieczenie i przechowanie światowego dorobku nauki i kultury, a przede wszystkim, jakimi metodami instytucje te mogą się posłużyć, aby pozostawić przyszłym pokoleniom świadectwo teraźniejszej intelektualnej działalności.

Do rozważań i działań w zakresie długoterminowego przechowywania dziedzictwa cyfrowego niejednokrotnie asumpt dawała publikacja J. Rothenberga [2], prekursora i światowej sławy specjalisty w dziedzinie emulacji[1]. W pracy tej autor roztacza wizję 2045 roku, kiedy to jego wnuki odnajdują na strychu odręcznie zapisany list z 1995 roku wraz z załączonym CD-ROM-em. W liście Rothenberg powiadamia je o tym, że na CD-ROM-ie jest zakodowana informacja o pozostawionym przez niego spadku, i jednocześnie wyjaśnia, w jaki sposób należy ją rozszyfrować. Odkrycie, choć wielce uszczęśliwiające, może się jednak okazać kłopotliwe, bowiem – jak przypuszcza autor – wnuki nie miały okazji widzieć wcześniej małego srebrnego krążka, chyba że w starych filmach. Nawet jeśli

[1] Jeff Rothenberg – czołowy informatyk w korporacji RAND w Kalifornii, autor wielu publikacji na temat długoterminowego archiwizowania publikacji elektronicznych, uznany za prekursora metody emulacji; jedną z jego prac jest *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation*, Washington 1999.

uda się im znaleźć odpowiedni czytnik CD, który dokona konwersji rozmieszczonych na krążku wgłębień na ciąg bitów, to kolejnym celem poszukiwań musi się stać program do ich prezentacji. Tak więc pytanie, czy wnuki Rothenberga skorzystają z nieoczekiwanego bogactwa, pozostaje otwarte.

Ten fikcyjny scenariusz uzmysławia zasadnicze słabości cyfrowego zapisu informacji w stosunku do zapisu tradycyjnego. Zdaniem Rothenberga, nawet za 50 i więcej lat napisany ręcznie list będzie można odczytać bezpośrednio, podczas gdy odczyt zapisów elektronicznych stanie się bardzo trudny lub niemożliwy z racji szybkiego tempa rozwoju w dziedzinie sprzętu i oprogramowania komputerowego. Wszelkie cyfrowe zapisy obecnego pokolenia, tak chętnie i masowo dziś wykorzystywane, ulegną dużo szybszemu zniszczeniu niż te utrwalone na papierze. Zawartość mediów cyfrowych staje się niemożliwa do odtworzenia znacznie szybciej niż słowa zapisane dobrym tuszem na dobrym papierze. Do utraty danych cyfrowych dochodzi najczęściej nie tyle z powodu fizycznego zniszczenia nośnika, ile z racji wprowadzania wciąż nowych, niekompatybilnych z wcześniejszymi generacji nośników i dedykowanych im urządzeń służących do odczytu danych. Najlepszym tego przykładem jest zastąpienie dyskietki 8-calowej kolejnymi generacjami nośników danych cyfrowych.

Do długoterminowej archiwizacji konieczne jest przechowywanie wraz z dokumentem elektronicznym informacji o wymaganym do jego odczytu otoczeniu sprzętowo-programowym. Najczęściej informacje te zawiera dokument wydrukowany na papierze oraz dołączony do cyfrowego medium – dokładnie jak w przedstawionej wcześniej opowieści o tym, jak przewidujący i troskliwy dziadek pozostawia wnukom papierowy list z opisem sposobu odkodowania informacji z krążka. Zdaniem Rothenberga naiwne jest myślenie, że teraźniejszy sposób cyfrowego kodowania treści będzie dla przyszłego oprogramowania tak samo oczywisty i łatwy w odczycie. Technologie informacyjne wprowadzą rozwiązania, które prawdopodobnie nie będą ani kontynuacją dotychczasowych, ani nie powstaną na ich bazie. Będą to nowe, oryginalne produkty, umiejętnie wdrożone na rynek i rozpowszechnione. Dokładnie tak stało się w przypadku taśm magnetycznych, kaset wideo tudzież dyskietek magnetycznych, które z czasem zostały zastąpione dyskami optycznymi. Czy zatem w obliczu tych zmian nie należałoby się zatroszczyć o utrzymanie dostępu do treści zapisanych na nośnikach wychodzących z obiegu, zwłaszcza gdy treści te są świadectwem istotnej naukowej i kulturalnej działalności człowieka? Czy brak należytej dbałości o to nie doprowadzi aby do ich bezpowrotnej utraty?

Swoje obawy Rothenberg uzasadnia, przytaczając przykład sprawozdania amerykańskiej Izby Reprezentantów z 1990 roku. Otóż wyniki powszechnego spisu ludności USA z 1960 roku zapisano na taśmach magnetycznych w dostępnym wówczas formacie zapisu, który znacznie wcześniej, niż się spodziewano, został zastąpiony nowszym. Tylko nieliczne z zapisanych danych udało się po latach od-

czytać i przenieść na nowe media. Tak samo zagrożone są bardzo ważne dane ministerstwa zdrowia USA, listy zaginionych i schwytanych podczas wojny w Wietnamie, protokoły z niepowtarzalnych eksperymentów agencji NASA. Bez końca można wymieniać przykłady unikatowych dokumentów cyfrowych zagrożonych utratą tylko dlatego, że z braku świadomości bądź stosownej wiedzy nie zrobiono nic dla ich długoterminowej archiwizacji.

2. Długoterminowa archiwizacja publikacji elektronicznych – wyjaśnienie pojęcia

Termin „archiwizacja publikacji elektronicznych" (ang. *long-term archiving*, niem. *langfristige Archivierung*) utożsamiany jest przede wszystkim z długoterminową ochroną, długoterminowym zabezpieczaniem i przechowywaniem dokumentów elektronicznych (ang. *long-term protection*, niem. *langfristige Erhaltung*, *dauerhafte Sicherung*), przy czym słowo „długoterminowy" należy tu rozumieć jako 'nieograniczony w czasie' lub 'możliwie najbardziej odległy w przyszłości'.

Długoterminowa archiwizacja powinna zapewnić użyteczność publikacji elektronicznych, czyli dostęp do nich i możliwość efektywnego korzystania z nich obecnie i w przyszłości. Efektywne korzystanie z publikacji elektronicznej jest możliwe wówczas, gdy użytkownik ma pewność, że treści, które czyta, ogląda, których słucha i na które powołuje się w swych opracowaniach, są autentyczne i niezafałszowane, tzn. że pochodzą od ich autorów i od dnia opublikowania nie zostały zmienione. Zapewnienie autentyczności i integralności publikacji elektronicznych jest podstawowym celem archiwizacji [3].

H. Liegmann[2] [4] zauważa, że warunkiem koniecznym zapewnienia użyteczności publikacji elektronicznych jest utrzymanie ich substancji. Pod pojęciem substancji publikacji elektronicznej autor rozumie ciąg bitów (kod zero-jedynkowy), zapisany na medium elektronicznym. Utrzymanie substancji publikacji elektronicznych jest uzależnione od dwóch zasadniczych czynników: ograniczonej trwałości mediów elektronicznych oraz szybkiego postępu technologicznego. Istotne znaczenie dla właściwego przebiegu archiwizacji ma przestrzeganie określonych zasad postępowania z publikacjami zapisanymi w formie cyfrowej, takich jak przestrzeganie instrukcji

[2] Hans Liegmann – pracownik Niemieckiej Biblioteki Narodowej we Frankfurcie nad Menem, specjalista w zakresie prac nad utrzymaniem długotrwałego i stabilnego dostępu do publikacji elektronicznych, przedstawiciel Biblioteki Niemieckiej w światowym zespole fachowców opracowujących strategię postępowania z dokumentami elektronicznymi, autor istotnych publikacji na temat długoterminowej archiwizacji publikacji elektronicznych w bibliotekach i archiwach.

dotyczących trwałości mediów elektronicznych, na których są zapisane publikacje, oraz kontrolowanie w ustalonych odstępach czasu, czy cyfrowy zapis na poziomie kodu zero-jedynkowego może zostać odczytany. Względy bezpieczeństwa dyktują, aby przed upływem pesymistycznie określonej granicy trwałości nośnika przekopiować zapisane na nim dane na nowy nośnik tego samego typu, np. z dyskietki na dyskietkę. Zabieg taki określany jest jako odświeżenie nośnika (ang. *refreshing*, niem. *Wiederauffrischen*). Natomiast w przypadku gdy nośnik przestaje być powszechnie stosowany i zastępuje go nowa generacja, treść dokumentu należy przekopiować na nośnik nowej generacji, np. z dyskietki 3,5-calowej na płytę CD-ROM lub DVD. Zabieg ten można nazwać zmianą generacji nośnika (ang. *reformating*, niem. *Wechsel der Tragergeneration*).

Kolejnym ważnym faktem, na który należy zwrócić uwagę, jest to, że trwałość medium elektronicznego jest zwykle dłuższa niż dostępność sprzętu i oprogramowania, potrzebnych do odczytu zapisanych na nim danych. Niezbędne jest zatem stałe obserwowanie zmian zachodzących w technologii i odpowiednio wczesne reagowanie na te zmiany. W anglojęzycznym piśmiennictwie przedmiotu proces ten nazwano *technology watch*, a w opracowaniach niemieckojęzycznych określono go jako *Frühwarnsystem*.

Samo zabezpieczenie substancji publikacji elektronicznej zapewniłoby użytkownikom jedynie dostęp do kodu zero-jedynkowego. Potrzebne są ponadto odpowiedni sprzęt i oprogramowanie, które umożliwią odczytanie zakodowanej w postaci zer i jedynek treści publikacji. Z racji tego, że w historii publikowania elektronicznego znajdowały zastosowanie różne platformy programowo-sprzętowe, w bibliotekach można znaleźć publikacje elektroniczne, których odczyt z nośnika i odszyfrowanie za pomocą aktualnego sprzętu i oprogramowania są dziś utrudnione, a często nawet niemożliwe. Przykładem mogą być publikacje zapisane na dyskietkach 5,25-calowych. Ich odczyt jest możliwy jedynie przy użyciu stacji dysków, które kilka lat temu wyszły z użycia. Nawet jeśli uda się zdobyć odpowiednią stację dysków, kolejną, trudną do pokonania barierą, może się stać dostępność platformy programowo-sprzętowej, niezbędnej do zdekodowania treści publikacji. Pomocne w takiej sytuacji okazuje się „zachowane technologii" [1]. Przechowywanie w bibliotekach sprzętu i oprogramowania, które wyszły z powszechnego użycia i wykorzystywane są jedynie w celu odczytywania publikacji zapisanych w formatach specyficznych dla tych platform, jest jedną z możliwych metod długoterminowej archiwizacji publikacji elektronicznych. Jednak tworzenie tzw. muzeów komputerowych nie spotyka się z uznaniem ze strony fachowców. Wypowiadając się na temat istotnych metod archiwizacji publikacji elektronicznych, wybierają oni migrowanie danych cyfrowych i emulowanie systemów [1].

3. Migracja i emulacja jako metody długoterminowej archiwizacji publikacji elektronicznych

Jedną z proponowanych metod długoterminowego utrzymywania użyteczności publikacji elektronicznych jest migracja, określana też jako konwersja dokumentu z oryginalnego formatu do nowszego, gdy format oryginalny staje się przestarzały i wychodzi z użycia [7, 8, 11]. Istota migracji polega na sukcesywnym przenoszeniu danych ze starszych, wychodzących z użycia formatów do formatów nowej generacji. Elektroniczny obiekt powinien zostać tak zmodyfikowany przez działania zewnętrzne, aby mógł być używany w zmienionym otoczeniu systemowym bez utraty danych treściowych i strukturalnych. Migracja danych do nowych warunków systemowych często wyklucza użycie publikacji w warunkach pierwotnych [4]. Konwersja danych może powodować powstanie przekłamań i odstępstw od oryginalnej wersji dokumentu – w wyglądzie zewnętrznym dokumentu, strukturze danych, interaktywnym zachowaniu, a nawet treści. Ponadto, jeśli kolejnej konwersji dokonuje się na podstawie rezultatu poprzedniej konwersji, to ryzyko przekłamań wzrasta, tym bardziej że nie ma już pierwotnego oryginalnego obiektu cyfrowego. Zatem jedynym sposobem uniknięcia przekłamań i odstępstw kolejnych wersji dokumentów od oryginałów jest użytkowanie dokumentów w ich oryginalnej aplikacji, czyli wyemulowanie ich pierwotnego otoczenia programowego.

W terminologii specjalistycznej pojęcia „emulować" używa się na określenie procesu naśladowania, symulowania, a także imitowania zachowań określonego sprzętu i oprogramowania [3, 5]. W komputerowej encyklopedii Microsoftu [6] termin ten jest definiowany jako proces imitowania przez komputer, urządzenie lub program funkcji, które spełnia inny komputer, urządzenie lub program. Metoda emulacji polega na tworzeniu programów emulujących starsze platformy programowo-sprzętowe na platformach aktualnie wykorzystywanych [1]. Zadaniem programów emulujących, nazywanych emulatorami, jest możliwie dokładne symulowanie architektury systemu, tak by różnica pomiędzy oryginalnym oraz naśladowanym systemem była niezauważalna. W przypadku dokumentów cyfrowych emulacja oznacza proces reprodukcji ich pierwotnego fenotypu [7].

Metoda emulacji stosowana jest głównie w przypadku publikacji elektronicznych, których treść i program ją prezentujący są ze sobą nierozerwalnie powiązane. Często zdarza się, że aplikacja stworzona dla określonego systemu operacyjnego jest z nim tak powiązana, że jej późniejsze przełożenie i zastosowanie w innych warunkach systemowych staje się niemożliwe. Zachodzi wówczas konieczność emulowania oryginalnego środowiska programowo-sprzętowego [4].

Emulowanie można więc postrzegać jako migrację nie danych cyfrowych, lecz otoczenia ich odczytu.

W publikacjach dotyczących omawianego zagadnienia [7, 8] emulację opisuje się jako metodę polegającą na „zmuszaniu" przyszłych technologii do funkcjonowania w taki sposób jak oryginalne środowisko zachowanego obiektu, co ma pozwolić na jego prezentację w pierwotnej postaci i na podstawie oryginalnego strumienia danych. Jednocześnie zwraca się uwagę na różnicę pomiędzy emulacją otoczenia sprzętowego a emulacją otoczenia programowego. Otóż za bardziej odpowiednią uznawana jest ta pierwsza, co wynika z stąd, że specyfikacje sprzętu mogą się okazać łatwiejsze do zdefiniowania niż specyfikacje oprogramowania. Poza tym emulacja platformy sprzętowej może być bardzo elastyczna i umożliwiać tym samym odtworzenie wielu systemów i prezentację różnych obiektów cyfrowych. Natomiast jako rozwiązanie alternatywne wymieniana jest emulacja pewnych aplikacji lub ich zachowań. Wadę takiego rozwiązania stanowi konieczność opracowania indywidualnego emulatora dla każdej aplikacji.

Informatycy [7] wymieniają trzy alternatywne elementy mogące stanowić przedmiot emulacji, są to: platforma sprzętowa, czyli komputer, platforma sprzętowa wraz z systemem operacyjnym, wreszcie kompletne otoczenie odczytu dokumentu – platforma sprzętowa wraz z systemem operacyjnym oraz programem umożliwiającym prezentację dokumentu. O tym, co będzie przedmiotem emulacji, powinno się decydować na etapie planowania strategii długoterminowej archiwizacji. Jeśli emulowany ma być sam komputer, to zachodzi konieczność zachowania dokumentu elektronicznego wraz z odpowiednim programem prezentującym go oraz systemem operacyjnym; jeśli emulacji mają podlegać komputer i system operacyjny, archiwizuje się dokument wraz z programem go prezentującym. Wydawałoby się, że rozwiązaniem optymalnym jest odtworzenie w procesie emulacji kompletnego otoczenia odczytu dokumentu elektronicznego. Informatycy tłumaczą jednak, że jest to złudne, ponieważ zwiększenie liczby kombinacji: komputer – system operacyjny – program umożliwiający prezentację dokumentu pociąga za sobą potrzebę tworzenia wielu emulatorów, tymczasem należy dążyć do minimalizacji liczby i złożoności narzędzi emulujących. Pomimo intensywnych badań i starań informatyków nadal problemem jest precyzyjne i kompletne odtworzenie działania kompleksowych programów. Dużo łatwiej jest odtworzyć działanie sprzętu, a zatem najbardziej zasadna jest alternatywa pierwsza, czyli emulacja samego komputera i archiwizowanie oprócz dokumentu elektronicznego, także stosownego systemu operacyjnego i programu umożliwiającego prezentację danego dokumentu.

Omawiając emulację, należałoby zwrócić też uwagę na jej wady i zalety jako metody długoterminowej archiwizacji obiektów cyfrowych. Potencjalną zaletę emulacji stanowi to, że jest ona znaną techniką informatyczną i wypracowała już emulatory różnych platform i systemów: od najstarszych, tworzonych przez entu-

zjastów, do systemów nowoczesnych, tworzonych w celach komercyjnych, służących do testowania i uruchamiania oprogramowania na różnych platformach. Przy możliwie najszerszym zastosowaniu tej metody emulacja umożliwiałaby odtworzenie pełnej funkcjonalności wielu obiektów cyfrowych (w tym oprogramowania) na podstawie oryginalnego, niezmodyfikowanego strumienia danych w połączeniu z oryginalnym oprogramowaniem [8]. Emulacja pozwala zapewnić autentyczność dokumentów elektronicznych dzięki zachowywaniu w długim czasie niezmiennej struktury oryginalnego dokumentu oraz wiernemu odtwarzaniu przez programy emulujące działania pierwotnego sprzętu. Nie ma również ograniczeń w zakresie typów dokumentów; specjaliści twierdzą, że nawet dokumenty dynamiczne[3] [9, 10] mogą zostać długoterminowo zarchiwizowane.

Technika emulacji oferuje znacznie więcej niż tylko długoterminową dostępność elektronicznych publikacji. Jej dodatkowym atutem jest to, że pozwala na zachowanie na przyszłość dowolnych systemów: sprawdza się na przykład w sektorze finansów i bankowości, gdzie umożliwia wieloletnie przechowywanie i odczyt dokumentów.

Wadę emulacji stanowi natomiast to, że jest ona metodą skomplikowaną technicznie, wymaga dużych nakładów pracy i specjalistycznej wiedzy, co pociąga za sobą poważne koszty. Jako metoda długoterminowej ochrony obiektów cyfrowych wymaga wciąż wielu badań. Konieczne jest systematyczne prowadzenie eksperymentów potwierdzających możliwość zastosowania tej metody do odczytu określonych typów publikacji elektronicznych. Jej wykorzystanie w przyszłości może być znacznie utrudnione, a w przypadku emulacji kompletnego środowiska odczytu i prezentacji dokumentu (sprzętu, systemu i oprogramowania) – praktycznie niemożliwe z racji nieodpowiedniej dokumentacji współczesnego oprogramowania, tudzież stosowania niestandardowych formatów. Niemożliwa wydaje się także emulacja wszystkich funkcji systemu lub aplikacji, a to z uwagi na wzrost złożoności systemów. Wreszcie, wraz ze zmianami przyszłych technologii i platform, również emulatory będą wymagały konwersji lub wyemulowania ich własnych środowisk w nowych systemach, co oznacza nakładanie się wielu warstw emulatorów [8].

[3] Wśród publikacji elektronicznych można wyróżnić m.in. publikacje statyczne i dynamiczne (*static and dynamic publications*). Statyczność publikacji elektronicznych polega na tym, że dokładnie ustalona treść i struktura dokumentu nie ulegają żadnym zmianom podczas dalszego użytkowania. Przykładem statycznych publikacji elektronicznych są monografie opublikowane w formie elektronicznej. Dynamiczność natomiast zaznacza się przez to, że treść publikacji elektronicznej i forma jej prezentacji są ustalane w trakcie ich użytkowania („w locie" – ang. *on the fly*). Przykładem dynamicznych publikacji elektronicznych są systemy bazodanowe. Ich treść stanowią generowane „w locie" rezultaty różnych zapytań wyszukiwawczych. Jednak to nie rezultat zapytania, lecz system bazodanowy jest obiektem archiwizacji.

Długoterminowe użytkowanie sprzętu i oprogramowania wymaga także regulacji prawnych. Symulowane przez emulatory systemy objęte są zwykle ochroną autorsko-prawną, zatem konieczne jest pozyskanie stosownych licencji. Problematyczne może się również okazać użytkowanie publikacji elektronicznych w dalekiej przyszłości z powodu całkiem realnych zmian, jakie mogą nastąpić w zakresie standardowych elementów obsługi sprzętu komputerowego. Jest bowiem wielce prawdopodobne, że klawiatura i mysz wyjdą z użycia, prawdopodobnie zmienią się także formy komunikowania się człowieka z maszyną [7].

4. Wnioski

Technologia emulacji, która umożliwia odtwarzanie oryginalnego technicznego otoczenia obiektu cyfrowego przy użyciu aktualnej technologii, jest traktowana jako strategia dla permanentnego dostępu do cyfrowego materiału [2]. Metoda ta wciąż jeszcze wymaga wielu istotnych badań i testów, potwierdzających jej pełną użyteczność w przypadku elektronicznych dokumentów bibliotecznych, archiwalnych tudzież muzealnych. Trudność polega na tym, że kierunek rozwoju i postęp w zakresie tworzonych narzędzi emulujących nie są podyktowane potrzebami instytucji pamięci, lecz zależą od potrzeb firm informatycznych. Badanie potrzeb bibliotek lub archiwów i opracowywanie rozwiązań wychodzących im naprzeciw to kosztowne inwestycje. Mogą sobie na nie pozwolić jedynie instytucje w bogatych krajach.

Istotnym warunkiem powodzenia przeprowadzanych w przyszłości emulacji wydaje się zapewnienie dużo większych możliwości (głównie wydajności) komputerów, na których trzeba będzie emulować obecne systemy. Warunek ten, z racji szybkiego rozwoju technologicznego i stale rosnących parametrów sprzętu, zostanie zapewne spełniony. Zatem można przyjąć, że dzięki emulacji realne staje się przechowanie dla przyszłych pokoleń autentycznych i integralnych dokumentów elektronicznych.

Konkludując, należy stwierdzić, że emulacja jest rekomendowana jako metoda reprodukcji pierwotnej, przestarzałej platformy na platformach nowszych i widzi się w niej najbardziej odpowiedni sposób prezentacji dokumentów cyfrowych w ich oryginalnym otoczeniu programowym w dalekiej przyszłości. Emulacja umożliwia przedstawienie oryginalnego dokumentu cyfrowego wraz z jego treścią, szatą graficzną, interfejsem oraz pozwala zachować jego tak zwany *look-and-feel*. Rezultaty przeprowadzanych eksperymentów wskazują, że emulacja może być zasadniczą metodą odczytu bibliotecznych, archiwalnych i muzealnych obiektów cyfrowych w przyszłości, pod warunkiem że stosowne emulatory przestarzałych platform komputerowych będą mogły działać na przyszłych platformach.

Specjaliści [2] zwracają uwagę na konieczność tworzenia technicznych specyfikacji, zawierających szczegółowy opis wszystkich istotnych cech platform sprzętowych w celu ich odtworzenia na przyszłych platformach. Dodatkowo należy dążyć do opracowywania takich technicznych rozwiązań, które umożliwią hosting potrzebnych emulatorów na przyszłych platformach przy minimalnym nakładzie starań.

Niezbędne jest również tworzenie metadanych opisujących dokument elektroniczny i przyporządkowujących go do odpowiedniego oprogramowania oraz emulatorów umożliwiających jego odczyt w przyszłości. Bezwzględnie potrzebne jest także zidentyfikowanie kryteriów autentyczności poszczególnych dokumentów cyfrowych i poprawności jej testowania w celu wprowadzenia mechanizmów oceny efektywności procesu długoterminowej archiwizacji bazującej na metodzie emulacji i tym samym skuteczności innych metod archiwizacji. Zalecane jest cykliczne wykonywanie testów i eksperymentów dotyczących emulacji w przyszłych badaniach nad jej potencjałem jako metody długoterminowej ochrony użyteczności bibliotecznych zasobów cyfrowych. Pozwoli to na jej udoskonalenie i umożliwi wzrost jej efektywności. Jeśli podczas serii takich eksperymentów nie wystąpią nieprzewidziane efekty, będzie to oznaczać, że problem długoterminowej ochrony danych cyfrowych można rozwiązywać właśnie przy zastosowaniu emulacji.

Bibliografia

[1] Czermiński J.B., *Cyfrowe środowisko współczesnej biblioteki*, Gdańsk 2002.

[2] Rothenberg J., *Ensuring the Longevity of Digital Documents*, „Scientific American" 1995, Vol. 272, No. 1.

[3] *Attributes of a Trusted Digital Repository: Meeting the Needs of Research Resources. RLG-OCLC Report*, Mountain View 2001, http://www.rlg.org/longterm/attributes01.pdf, [dostęp: 20.04.2007].

[4] Liegmann H., *Langzeitverfügbarkeit digitaler Publikationen*, [w:] *Bibliotheken – Portale zum globalen Wissen. 91. Deutscher Bibliothekartag in Bielefeld 2001*, Frankfurt am Main 2001.

[5] *Lexikon Informatik und Datenverarbeitung*, München–Wien 1997.

[6] Woodcock J., Aiken P. i in. (red.), *Microsoft. Encyklopedia komputerowa*, Warszawa 2002.

[7] Borghoff U.M., *Langzeitarchivierung. Methoden zur Erhaltung digitaler Dokumente*, Heidelberg 2003.

[8] National Library of Australia (oprac.), *Ochrona dziedzictwa cyfrowego. Zalecenia*, Warszawa 2003.

[9] Steenbakkers J., *Setting up a Deposit System for Electronic Publications. The NEDLIB Guidelines*, Amsterdam 2000.

[10] Januszko-Szakiel A., *Archiwizacja publikacji elektronicznych jako wyzwanie dla bibliotek – zarys problematyki*, „Biuletyn Biblioteki Jagiellońskiej" 2003.
[11] Oltmans E., *A Comparison between Migration and Emulation in Terms of Costs*, http://www.rlg.org/en/page.php?Page_ID=20571, [dostęp: 21.04.2007].