

Grzegorz DEC, Bartosz JĘDRZEJEC, Wojciech RZĄSA
Politechnika Rzeszowska, Katedra Informatyki i Automatyki

KOLOROWANA SIEĆ PETRIEGO JAKO MODEL SYSTEMU PODEJMOWANIA DECYZJI KREDYTOWEJ

Streszczenie. W artykule opisano model formalny i implementację systemu ekspertowego (SE) do podejmowania decyzji kredytowych. SE projektowany jest jako zbiór powiązanych tabel decyzyjnych (TD). Jako model tego SE zastosowano kolorowaną sieć Petriego (CPN), na którą odwzorowano strukturę SE oraz sposób wyznaczania wartości przez TD. Utworzono moduł podejmowania decyzji kredytowych, który jest częścią systemu informatycznego przeznaczonego dla bankowości.

Słowa kluczowe: kolorowana sieć Petriego, system ekspertowy

COLORED PETRI NET AS A MODEL OF A SYSTEM FOR SUPPORTING CREDIT DECISIONS

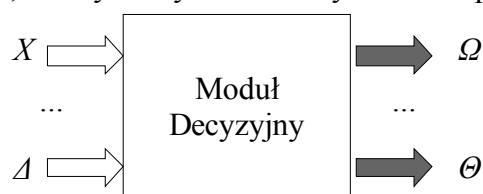
Abstract. The paper describes a formal model and implementation of an expert system (ES) for a credit decision making. The ES is designed as a set of related decision tables (DT). The Colored Petri Net (CPN) is adopted as a model of the ES. The structure and behavior of the ES is mapped into the CPN. A software module for a credit decision making is developed and deployed to a computer system for banking.

Keywords: colored Petri net, expert systems

1. Motywacja

W artykule opisano model formalny i implementację systemu ekspertowego (SE) służącego do podejmowania decyzji kredytowych. Motywacją do podjęcia prac było zapotrzebowanie zgłoszone przez firmę Assec Poland SA na rozbudowanie systemu informatycznego Sales Agent o dodatkowy komponent – moduł decyzyjny, używany w procesie przyznawania kredytu, w celu automatycznego wyznaczenia parametrów pożyczki.

Moduł decyzyjny (MD) jest systemem informatycznym, który na podstawie parametrów wejściowych wyznacza parametry kredytu (rys. 1). Parametrami wejściowymi są wielkości charakteryzujące klienta bankowego, takie jak np. wielkość dochodu, miejsce zatrudnienia. Wartościami wyjściowymi MD są wielkości charakteryzujące udzielany kredyt, takie jak maksymalna kwota kredytu, maksymalny okres kredytowania itp.



Rys. Moduł decyzyjny jako system informatyczny

1.

Fig. 1. The decision module as an information system

1.1. Opis problemu

Zgodnie z wymaganiami zleceniodawcy MD projektowany jest przez analityka bankowego, który definiuje strukturę modułu jako zbiór powiązanych ze sobą tabel decyzyjnych oraz jego dynamikę, wypełniając tabele wartościami (rys. 2). Każda tabela posiada określone parametry wejściowe i wyjściowe. Parametrami wejściowymi tabeli mogą być parametry modułu lub wartości wyznaczone przez inne tabele. Każdy z parametrów posiada pewien typ – zbiór wartości, które mogą być przypisane do parametru. Tabele mają dowolną ilość parametrów wejściowych i wyjściowych.

Specyfikacja systemu wymagała, aby interfejs użytkownika MD skonstruowany był w taki sposób, żeby niezależnie od zastosowanego modelu systemu ekspertowego definiowanie go odbywało się za pomocą tabel decyzyjnych.

W ogólnym przypadku zarówno MD jak i poszczególne tabele decyzyjne są funkcjami postaci:

$$Y = F(X),$$

gdzie $X = \{X_1, \dots, X_n\}$ i $Y = \{Y_1, \dots, Y_m\}$ są wektorami zawierającymi zbiory określoności parametrów wejściowych i wyjściowych MD lub tabeli (typy parametrów). Na przykład tabela 4 z rys. 2 jest odwzorowaniem postaci:

$$Z \times C \times A \rightarrow \Omega,$$

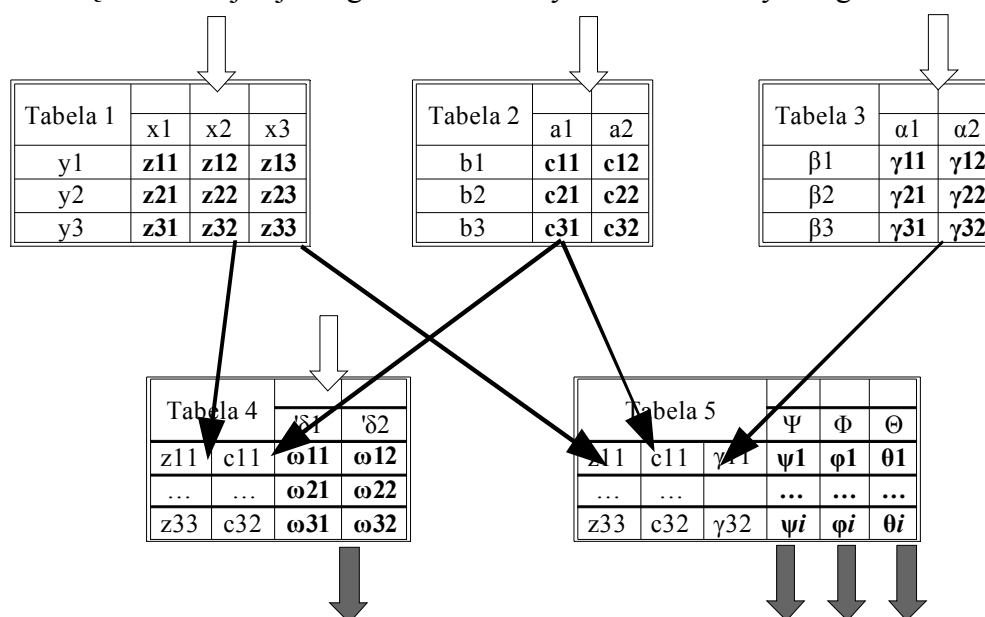
gdzie:

$$Z = \{z_{11}, \dots, z_{33}\}, C = \{c_{11}, \dots, c_{32}\}, A = \{\delta_1, \delta_2\}, \Omega = \{\omega_{11}, \dots, \omega_{32}\}.$$

Na podstawie analizy specyfikacji wymagań MD dokonano klasyfikacji typów parametrów oraz odwzorowań realizowanych przez tabele decyzyjne. Typy parametrów MD i tabel decyzyjnych sklasyfikowane zostały następująco:

- typ 1: zbiór liczb rzeczywistych \mathcal{R} ,
- typ 2: zbiór, np. dochód klienta może przyjmować wartości $\{\text{niski}, \text{średni}, \text{wysoki}\}$,
- typ 3: zbiór liczb rzeczywistych podzielony na przedziały, np. $(-\infty; x_1>$, $(x_1; x_2>$, ..., $(x_n; +\infty)$.

Potrzeba zastosowania ostatniego z wymienionych typów związana była z koniecznością dostosowania się do interfejsu jednego z modułów systemu informatycznego.



Rys. 2. Tabele decyzyjne tworzące przykład modułu decyzyjnego

Fig. 2. Decision tables forming an example decision module

Odwzorowania realizowane przez tabele decyzyjne sklasyfikowane zostały na dwa typy:

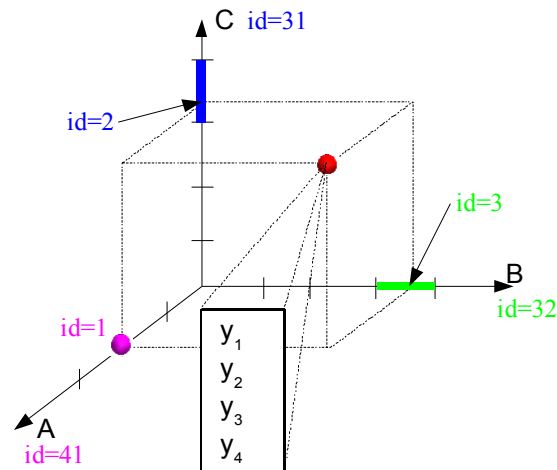
- Typ 1: odwzorowanie opisane wyrażeniem

$$\mathbf{Y} = c_1 \cdot A \cdot X + c_2 \cdot B,$$
 gdzie A , X , B są parametrami określonymi na zbiorze liczb rzeczywistych, a $\mathbf{Y} = \{Y_1, \dots, Y_m\}$ jest wektorem parametrów wyjściowych tabeli decyzyjnej, z których każdy jest również określony na zbiorze \mathcal{R} ,
- Typ 2: odwzorowanie przestrzeni n – wymiarowej w m – wymiarową, gdzie dziedzinę stanowi iloczyn kartezyjański zbiorów typów 2 lub 3, a przeciwdziedziną jest iloczyn kartezyjański zbiorów typów 1 lub 2 (rys. 3).

Rozpatrywany system ekspercki, zaprojektowany jako zbiór tabel decyzyjnych, jest równoważny systemowi regułowemu [1] [2]. Tabele decyzyjne modułu decyzyjnego można więc przedstawić jako zbiór implikacji postaci:

Jeżeli x jest x_1 i y jest y_1 i ... to z jest z_1 .

W punkcie 2 zostanie pokazane, że tabela decyzyjna modelująca odwzorowanie typu 1 (wyrażenie arytmetyczne) również może być zastąpiona implikacją.



Rys. Przykład odwzorowania $A \times B \times C \rightarrow \mathbb{R}^4$
3. realizowanego przez tabelę decyzyjną

Fig. 3. An example mapping $A \times B \times C \rightarrow \mathbb{R}^4$ realized
by a decision table

1.2. Przegląd dostępnych rozwiązań

Na rynku dostępne są gotowe moduły systemów eksperckich, realizujące wnioskowanie na podstawie zadanych przesłanek. Omówione zostaną produkty, które mogłyby znaleźć zastosowanie w podobnych przypadkach.

W serwerze aplikacji JBoss [3] dostępny jest moduł o nazwie JBoss Rules, który implementuje wnioskowanie na podstawie reguł w oparciu o algorytm RETE [4]. Systemy Jess [5] i SOAR [6] są bibliotekami programistycznymi, które implementują funkcje systemów eksperckich. Mogą one być dołączone dynamicznie do istniejących aplikacji. Wymienione komponenty charakteryzują się wspólnymi cechami:

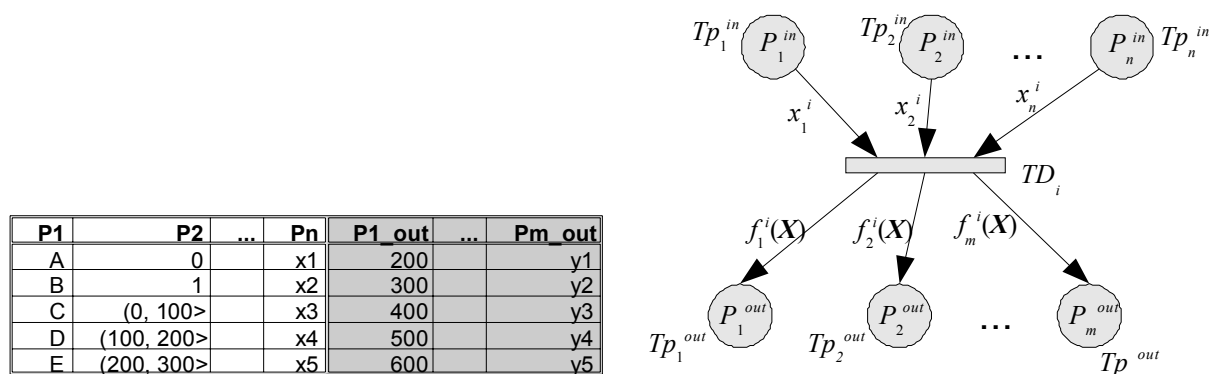
- system ekspercki specyfikowany jest jako zbiór reguł postaci „jeżeli P to Q”, gdzie P i Q są zdaniami logicznymi,
- oparte są na logice dwuwartościowej,
- nie dają możliwości wyznaczania wartości operacji arytmetycznych postaci

$$y = c_1 \cdot A \cdot X + c_2 \cdot B.$$

Przy zastosowaniu wspomnianych bibliotek programowych liczba reguł równa jest liczbie wierszy w tabelach decyzyjnych. Właściwość ta sprawia, że model matematyczny używany do wyznaczania wartości tabel jest trudny do analizy z punktu widzenia analityka bankowego.

Cechą, która uniemożliwiła zastosowanie wspomnianych rozwiązań, jest brak bezpośredniego wsparcia dla traktowania tabel decyzyjnych wyznaczających wartości wyrażeń arytmetycznych jako reguł systemu ekspertowego. Realizacja zadania przy użyciu

ww. narzędzi wprowadziłyby do modelu niespójność i komplikacje związane z koniecznością specjalnego traktowania wspomnianych tabel decyzyjnych.



Rys. 4. Tranzycja CPN modelująca tabelę decyzyjną

Fig. 4. A transition of the CPN modeling a decision table

2. Kolorowana sieć Petriego jako model modułu decyzyjnego

Kolorowane sieci Petriego (Colored Petri Net - CPN) [9] znajdują zastosowanie jako modele formalne regułowych systemów eksperckich, opartych na logice dwuwartościowej [7] [8] jak i rozmytej [10]. Systemy eksperckie oparte na logice rozmytej bardzo często utożsamiane są z rozmytymi sieciami Petriego [11][12][13], dedykowanymi do rozwiązywania specyficznych problemów. W przypadku rozpatrywanym w tej pracy logika rozmyta nie znalazła zastosowania z przyczyn leżących po stronie odbiorców systemu.

Wspólną cechą występującą w literaturze, która opisuje związek systemów regułowych z sieciami Petriego, jest powiązanie reguł SE z tranzycjami sieci oraz zdań logicznych z poprzedników i następników implikacji z miejscami sieci. W celu opracowania modelu formalnego modułu decyzyjnego autorzy pracy zastosowali klasyczną jensenowską kolorowaną sieć Petriego, przyjmując następujące założenia:

- każdej tabeli decyzyjnej odpowiada tranzycja w CPN,
- każdy parametr wejściowy lub wyjściowy tabeli decyzyjnej jest miejscem CPN, połączonym odpowiednim łukiem z tranzycją,
- typy parametrów tabel decyzyjnych odwzorowane są na kolory tokenów CPN,
- sieć jest bezkonfliktowa, zachowawcza, o przepustowości łuków i pojemności miejsc równej 1.

Rysunek 4 pokazuje fragment CPN modelujący tabelę decyzyjną TD_i o n parametrach wejściowych i m parametrach wyjściowych. Z każdym miejscem P_i^{in} związany jest typ Tp_i^{in} . Z łukami związane są wyrażenia łuków. Wyrażenia łuków wejściowych używane są

w przypadku konwersji typów parametrów z liczby rzeczywistej na przedziały. Wyrażenia łuków wyjściowych są funkcjami utworzonymi na podstawie tabeli decyzyjnej.

Na przykładzie z rysunku 4 mamy:

$$Tp_1^{in} = \{A, B, C, D, E, F\}$$

$$Tp_2^{in} = \{0, 1, (0, 100>, (100, 200>, (200, 300> \}$$

$$Tp_n^{in} = \{x_1, x_2, x_3, x_4, x_5\}$$

$$Tp_1^{out} = \mathbb{R}$$

$$Tp_m^{out} = \{y_1, y_2, y_3, y_4, y_5\}$$

$$X \subset Tp_1^{in} \times Tp_2^{in} \times Tp_n^{in}$$

$$f_1(X): (Tp_1^{in} \times Tp_2^{in} \times Tp_n^{in}) \rightarrow Tp_1^{out}$$

$$f_m(X): (Tp_1^{in} \times Tp_2^{in} \times Tp_n^{in}) \rightarrow Tp_m^{out}$$

Dzięki opisanym założeniom możliwe jest zastosowanie podejścia analogicznego do pokazanego w pracy [12] i przedstawienie modułu decyzyjnego jako zbioru reguł postaci:

Jeżeli x jest X i ... i y jest Y to v jest V i ...,

gdzie:

- x, y i v są parametrami tabeli (zmiennymi lingwistycznymi) określonymi na dziedzinach odpowiednio $X = \{x_1, \dots, x_k\}$, $Y = \{y_1, \dots, y_l\}$ i $V = \{v_1, \dots, v_m\}$,
- X, Y, V są zbiorami rozmytymi, takimi że $X: X \rightarrow \{0, 1\}$, $Y: Y \rightarrow \{0, 1\}$, $V: V \rightarrow \{0, 1\}$ i $X(x)$ określa stopień przynależności x do X , $Y(y)$ określa stopień przynależności y do Y , $V(v)$ określa stopień przynależności v do V .

Implikacje określone są jako odwzorowania przypisane do łuków wyjściowych tranzycji CPN.

W przypadku reguł modelujących wyrażenia arytmetyczne (odwzorowanie realizowane przez tabele decyzyjne typu 2) wprowadzamy następującą regułę:

Jeżeli a jest A i x jest X i b jest B , to $y = c_1 \cdot a \cdot x + c_2 \cdot b$,

gdzie A, X, B są zbiorami rozmytymi zdefiniowanymi jak poprzednio, a parametry a, x, b oraz y określone są na zbiorze liczb rzeczywistych. Rysunek 5 pokazuje fragment CPN modelujący regułę wyznaczającą wartość wyrażenia arytmetycznego. Analityk projektuje wyrażenie łuku wyjściowego $f_T(X)$ określając przyporządkowania:

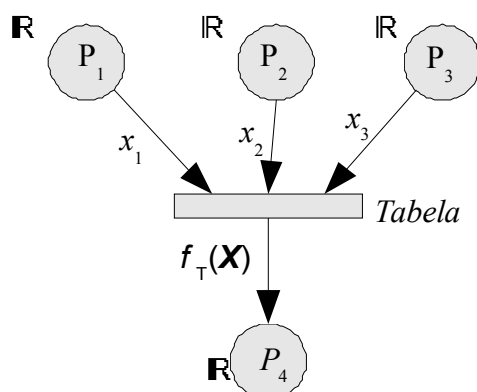
$$\{a, x, b\} \rightarrow \{P_1, P_2, P_3\},$$

$$\{0, -1, 1\} \rightarrow c_1,$$

$$\{0, -1, 1\} \rightarrow c_2.$$

Przyjęty model odwzorowuje każdą tabelę decyzyjną na jedną regułę (tranzycję na CPN), bez względu na typ odwzorowania realizowanego przez tabelę. Pozwala to na łatwą weryfikację poprawności działania systemu eksperckiego, gdyż model zawiera jedynie takie składniki (tranzycje i miejsca), jakie analityk wprowadził tworząc tabele decyzyjne.

Uniknięto w ten sposób wygenerowania znacznej ilości reguł, których pojawienie się w modelu byłoby niezrozumiałe dla analityka.

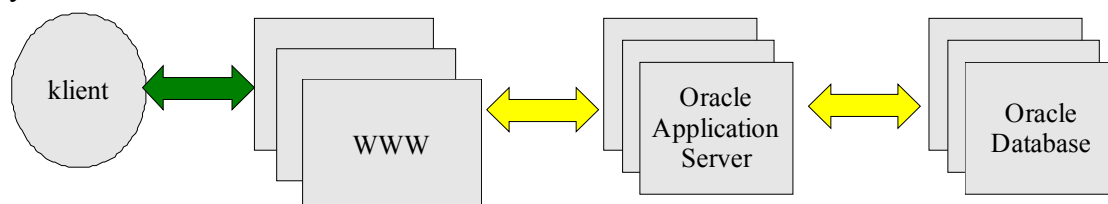


Rys. 5. Tranzycja CPN modelująca wyrażenie arytmetyczne
Fig. 5. A transition of the CPN modeling an arithmetic expression

3. Implementacja systemu

Moduł decyzyjny zaimplementowany został w technologii Java Enterprise Edition. System, którego składnikiem jest moduł, ma 3-warstwową architekturę typu klient – serwer z rozkładaniem obciążenia (rys. 6). Warstwa prezentacji wykonana jest w technologii Java Server Faces, logika biznesowa korzysta z komponentów sesyjnych Enterprise Java Beans 3.0, warstwa danych używa komponentów encyjnych EJB 3.0 oraz odwzorowania obiektowo – relacyjnego Toplink JPA. W relacyjnej bazie danych utworzono tabele przechowujące strukturę CPN, tj. miejsca, tranzycje, łuki, kolory tokenów. Funkcje realizowane przez poszczególne tabele decyzyjne implementowane są na poziomie serwera aplikacji w języku Java i odpowiednie obiekty zapisane są po procesie serializacji w bazie danych.

Dzięki analizie właściwości statycznych sieci Petriego moduł wykrywa konflikty w przesłankach i konkluzjach oraz reguły tworzące pętle, nie pozwalając analitykowi na wprowadzenie błędów strukturalnych do systemu. Użytkownik ma możliwość oglądnięcia grafu CPN, który rysowany jest automatycznie na podstawie struktury sieci zapisanej w bazie danych.



Rys. 6. Architektura systemu zawierającego moduł decyzyjny
Fig. 6. The architecture of the information system containing the decision module

4. Podsumowanie

W wyniku prac opisanych w artykule rozbudowano system informatyczny Sales Agent o dodatkowy komponent – moduł decyzyjny wyznaczający parametry kredytów. Przedstawione rozwiązanie charakteryzuje się następującymi cechami:

- zastosowano dobrze znany formalizm do opisu struktury i zachowania systemu ekspertowego,
- wyznaczanie wartości tabel decyzyjnych odbywa się poprzez symulację CPN,
- wyznaczenie właściwości statycznych systemu (konflikty, pętle) jest proste, dzięki analizie statyki sieci Petriego,
- model ma reprezentację graficzną za pomocą grafu sieci Petriego,
- liczba tranzycji CPN jest równa liczbie tabel decyzyjnych, a miejsca CPN odpowiadają parametrom tabel, co czyni analizę CPN intuicyjną dla analityków bankowych.

System ma możliwości dalszego rozwoju. Oczywiście jest, że prawidłowy model systemu powinien bazować na logice rozmytej, co uczyni proces podejmowania decyzji bardziej elastycznym. Aktualnie nie jest sprawdzana logiczna poprawność reguł decyzyjnych. Korzystając z bazy danych o spłatach kredytów możliwe jest uzupełnienie modułu decyzyjnego o komponent weryfikujący poprawność reguł utworzonych przez analityka.

BIBLIOGRAFIA

1. Vanthienen J., Wets G.: From decision tables to expert system shells. *Data & Knowledge Engineering*, Vol. 13 Iss. 3, s. 265÷282.
2. Ligęza A.: *Logical Foundations for Rule-Based Systems*. Wyd. AGH, Kraków 2005.
3. Fleury M., Reverbel F.: The JBoss Extensible Server. *Proc. of the ACM/IFIP/USENIX 2003 International Conference on Middleware*, Rio de Janeiro, Brazil 2003, s. 344÷373.
4. Forgy C.: *On the efficient implementation of production systems*. Ph.D. Thesis, Carnegie-Mellon University, 1979.
5. Friedman-Hill E.: *Jess in Action: Java Rule-Based Systems*. Greenwich, Manning Publications, 2003.
6. Laird J.E., Newell A., Rosenbloom P.S.: SOAR: an architecture for general intelligence. *Artificial Intelligence*. Vol. 33, No. 1 (Sep. 1987), s. 1÷64.
7. Shiu S.C.K., Liu J.N.K., Yeung D.S.: Modelling Hybrid Rule/Frame-based Expert Systems using Coloured Petri Nets. *Proc. of the 8th Int. Conf. on Industrial & Engineering Applications of AI and Expert Systems*, Melbourne, Australia 1995, s. 525÷531.

8. Chao-Shun C., Chia-Hung L., Hung-Ying T.: A rule-based expert system with colored Petri net models for distribution system service restoration. *IEEE Transactions on Power Systems*, Vol. 17, Iss. 4, s. 1073÷1080, 2002.
9. Jensen K.: *Coloured Petri Nets. Basic Concepts, Analysis Methods and Practical Use. Volume 1, Basic Concepts. Monographs in Theoretical Computer Science*, Springer-Verlag, 1997.
10. Zuleta L.E.L., Madrigal G.Z., Ovalle Carranza D.A.: Hybrid System based on Fuzzy Inference and Colored Petri Nets to Identify Electrical Fault Events in Real Time. *Proc. Electronics, Robotics and Automotive Mechanics Conference*, 25-28 Sept. 2007, CERMA 2007, s. 400÷405.
11. Looney C.G.: Fuzzy Petri nets for rule-based decisionmaking. *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 18, Issue 1, 1988, s. 178÷183.
12. Scarpelli H., Gomide, F., Yager R.R.: A reasoning algorithm for high-level fuzzy Petri nets. *IEEE Transactions on Fuzzy Systems*, Vol. 4, Issue 3, Aug 1996, s. 282÷294.
13. Dec G.: *Rozmyty system ekspertowy jako sprzętowy układ sterowania i kontroli*. Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych, praca doktorska, 2006.

Recenzenci: Dr hab. inż. Andrzej Kwiecień, prof. Pol. Śląskiej
Dr hab. inż. Mirosław Zaborowski

Wpłynęło do Redakcji 31 stycznia 2010 r.

Abstract

The paper describes a formal model and implementation of a decision module (DM) – an expert system for a credit decision making. The motivation for the work was a request from the Asseco Poland S.A. for improving the Sales Agent computer system.

The DM is designed as a set of related decision tables (fig. 2). Each decision table can be considered as a generic function, that performs specified mapping (fig. 3). Besides those, a table is allowed, that performs equation:

$$Y = c_1 \cdot A \cdot X + c_2 \cdot B$$

where A , B , X are parameters of the table.

The DM designed as a collection of tables is equivalent to a rule based system [1, 2]. Decision tables can be expressed as a set of implications:

If x is x_1 and y is y_1 and ... then z is z_1 .

However, the specification of the DM stated, that the end user should only see decision tables but no model behind them.

There are some solutions of the software modules for rule based expert system. None of them could be used because of complexity of decision tables.

The Colored Petri Nets (CPN) are common models for rule based expert systems. Usually implications are mapped to transitions of the CPN and logic sentences are mapped to places of the CPN. Authors of this paper proposed the same solution. Fig. 4 shows the dependency between a decision table and a CPN. Expressions of the outgoing arcs are used for implementation of the function that is performed by the decision table.

The decision module is implemented using Jave Enterprise Edition technology as a 3-tired Internet application (fig. 6). The presentation layer uses Java Server Faces, the business logic layer is a stateless session component EJB 3.0, and the data layer benefits from EJB 3.0 entities.

Adresy

Grzegorz DEC: Politechnika Rzeszowska, Katedra Informatyki i Automatyki, ul. W. Pola 2, 35-959 Rzeszów, Polska, gdec@prz-rzeszow.pl

Bartosz JĘDRZEJEC: Politechnika Rzeszowska, Katedra Informatyki i Automatyki, ul. W. Pola 2, 35-959 Rzeszów, Polska, bartoszj@prz-rzeszow.pl

Wojciech RZAŚA: Politechnika Rzeszowska, Katedra Informatyki i Automatyki, ul. W. Pola 2, 35-959 Rzeszów, Polska, wrzasa@prz-rzeszow.pl