

KIWIEL



**POLSKA AKADEMIA NAUK**  
**Instytut Badań Systemowych**

# **WSPOMAGANIE DECYZJI**

# **SYSTEMY EKSPERCKIE**

pod redakcją

**Romana Kulikowskiego i Lucyny Bogdan**

Warszawa 1995

# **WSPOMAGANIE DECYZJI**

## **SYSTEMY EKSPERCKIE**

pod redakcją

**Romana Kulikowskiego i Lucyny Bogdan**

Warszawa 1995

Wydano z wykorzystaniem dotacji  
KOMITETU BADAŃ NAUKOWYCH

Materiały konferencji: "Analiza Decyzyjna, Systemy Ekspertyczne, Zastosowania Systemów Komputerowych",  
Warszawa, 25-27 maja 1994r.

Komitet Programowy Konferencji:

Andrzej Ameljańczyk, Zdzisław Bubnicki, Wiesław Grudzewski, Olgierd Hryniewicz, Janusz Kacprzyk, Lech Kruś, Roman Kulikowski (przewodniczący), Kazimierz Mańczak, Ireneusz Nykowski, Zdzisław Pawlak, Roman Słowiński, Andrzej Straszak, Andrzej Weryński, Andrzej Wierzbicki.

Wykonano z oryginałów tekstowych dostarczonych przez autorów

© Instytut Badań Systemowych PAN, Warszawa 1995

ISBN 83-85847-85-5

# MECHANIZMY KONTROLI WSPÓŁPRACY ŹRÓDEŁ WIEDZY W SYSTEMIE EKSPERCKIM O ARCHITEKTURZE TABLICOWEJ

Artur Michalski\*

Instytut Informatyki Politechniki Poznańskiej

## Streszczenie

W pracy przedstawiono propozycję rozwiązania problemu sterowania w systemie eksperckim o architekturze tablicowej. Opisano główne elementy architektury, cykl sterowania oraz funkcjonowanie poszczególnych modułów sterujących.

## WSTĘP

W systemie eksperckim o architekturze tablicowej wiedza konieczna do rozwiązania problemu nie jest zgromadzona w jednym module, lecz podzielona na pewne niezależne fragmenty określane mianem *źródeł wiedzy*. Niezależność źródeł polega na tym, że mają one własne mechanizmy wnioskujące, mogą wykorzystywać różne formy reprezentacji wiedzy oraz nie dysponują żadną informacją o innych źródłach.

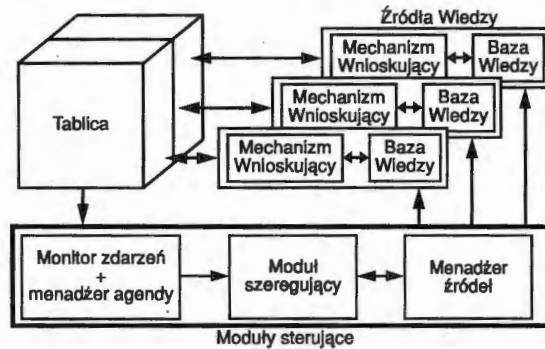
Współpraca i komunikacja między źródłami może się odbywać jedynie za pomocą struktury zwanej *tablicą*. Tablica jest rodzajem pamięci roboczej współdzielonej przez źródła, miejscem, gdzie przechowywany jest bieżący stan procesu rozwiązywania problemu.

Nad współpracą źródeł czuwają moduły sterujące. Ich zadaniem jest śledzić zmiany zachodzące w tablicy i podejmować decyzję o tym, jaka będzie następna akcja, tzn. gdzie, kiedy i jak zostanie użyta wiedza zawarta w źródłach [2].

---

\* Praca wykonana w ramach grantu KBN 3 P406 001 05 (43-0571)

Na rys.1 przedstawiono architekturę systemu tablicowego będącego tematem pracy. W pierwszym punkcie omówiono podstawowe struktury wchodzące w skład systemu. W następnych punktach opisano szczegółowo funkcjonowanie poszczególnych modułów sterujących



Rys. 1. Architektura tablicowego systemu eksperckiego

## PODSTAWOWE ELEMENTY ARCHITEKTURY

### TABLICA

Tablica ma charakter obiektowej bazy danych dostępnej dla wszystkich źródeł wiedzy. Nie jest podzielona na poziomy abstrakcji [3,4,5] ani inne mniejsze fragmenty, bowiem dzięki reprezentacji obiektowej projektant aplikacji ma pełną swobodę w organizowaniu jej struktury. W tym celu można korzystać z wszystkich mechanizmów charakterystycznych dla programowania obiektowego: dziedziczenia, hermetyzacji i polimorfizmu. Najmniejszą, elementarną jednostkę tablicy stanowi instancja klasy obiektów, opisana przez atrybuty i metody. Na rys. 2 pokazano przykładowy obiekt tablicy (bez metod).

CLASS	task
LABEL	"machine scheduling task"
ATTRIBUTE	NAME start_time
	TYPE numeric
	VALUE 0.0
	NAME end_time
	TYPE numeric
	VALUE 0.0
	NAME resource_type
	TYPE object

Rys. 2. Przykładowy obiekt tablicy

NAME	event_15
EVENT	assign_start
SUBJECT	task.start_time
KIND	add
AUTHOR	basic_scheduler
RATE	4

Rys. 3. Przykładowe zdarzenie

Obiekty zawarte w tablicy mogą reprezentować takie dane jak: dane wejściowe, wyniki pośrednie, rozwiązania częściowe i alternatywne, rozwiązania końcowe.

Utworzenie lub usunięcie obiektu bądź kilku obiektów lub tylko zmiana jego atrybutu może spowodować pojawienie się *zdarzenia*. Każde zdarzenie charakteryzuje się (rys. 3):

a) nazwą (pole NAME) - unikalny identyfikator do użytku wewnątrz systemu,



- b) klasą zdarzeń (pole EVENT) - zbiór zmian uznawanych za zdarzenie, definiowany przez projektanta aplikacji,
- c) miejscem wystąpienia (pole SUBJECT) - wskaźnik do obiektu i/lub atrybutu, gdzie wystąpiła zmiana,
- d) typem zmiany (pole KIND) - charakter jaki przyjęła zmiana (nowa wartość, modyfikacja wartości, usunięcie wartości),
- e) autorem zmiany (pole AUTHOR) - identyfikator źródła wiedzy, które dokonało zmiany,
- f) wagą zmiany (pole RATE) - ocena odzwierciedlająca znaczenie przypisywane zmianie przez inżyniera wiedzy (patrz dalej: oceny reguł w bazie wiedzy).

Zdarzenia kojarzy się ze źródłami wiedzy w celu precyzyjnego określenia sytuacji w tablicy, w których źródła mają włączać się do współpracy z innymi źródłami (patrz dalej: warunki aktywacji źródeł wiedzy).

### ŹRÓDŁA WIEDZY

Źródła wiedzy reprezentują niezależnych ekspertów, którzy biorą udział w rozwiązaniu problemu. Każde z nich obserwuje zmiany zachodzące w tablicy i próbuje przybliżyć system do ostatecznego rozwiązania. W tym celu w odpowiednich momentach, określanych jako *warunki aktywacji* (definiowanych przez inżyniera wiedzy odpowiadającego za konstrukcję danego źródła wiedzy), źródła podejmują próbę zastosowania posiadanej wiedzy zgodnie z posiadanym mechanizmem wnioskowania do zmiany bieżącego stanu zapamiętanego w tablicy.

---

```

KS          constraints_checker
TRIGGER    new_task, assign_start,
           assign_end, assign_resource
METHOD     forward
BASE       "c_check.kb"
IMPORTANCE 0.8
COMPLEXITY 27

```

---

Rys. 4. Przykładowy deskryptor źródła wiedzy

---

```

VARIABLE   CLASS task
           OBJECT task1, task2
           CLASS resources
           OBJECT res1, res2, res3

RULE       delay_start_time
IF         lt(task1.start_time, 10)
AND        eq(task1.resource_type, tester)
THEN       modify(task1.start_time, 21)
RATE      3

```

---

Rys. 5. Fragment przykładowej bazy wiedzy

Każde źródło składa się z dwóch części: deskryptora źródła i bazy wiedzy. W skład *deskryptora* wchodzi (rys. 4):

- a) nazwa (pole KS) - unikalny identyfikator źródła wiedzy,
- b) warunki aktywacji (pole TRIGGER) - lista zdarzeń, przy których źródło wiedzy ma się uaktywniać,
- c) metoda wnioskowania (pole METHOD) - niezależny mechanizm rozumowania (sterowany danymi lub sterowany celem) stosowany przez źródło,
- d) znaczenie źródła (pole IMPORTANCE) - sumaryczna ocena zmian wnoszonych przez źródło, wyznaczana na podstawie ocen wszystkich reguł (rys. 5 pole RATE) zawartych w bazie,
- e) złożoność bazy wiedzy (pole COMPLEXITY) - liczba wszystkich (różnych) warunków elementarnych zawartych w przesłankach reguł z bazy wiedzy.

*Baza wiedzy* (rys. 5) składa się z definicji obiektów wykorzystywanych przez źródło (pole VARIABLE) oraz ze zbioru reguł produkcji (pola RULE...) reprezentujących właściwą wiedzę przedmiotową, stosowaną przy rozwiązywaniu problemu. Użycie każdego obiektu w treści reguły musi być poprzedzone jego deklaracją w polu zmiennych. Reguły mają typową budowę: IF <warunek> THEN <akcja> oraz dodatkowo przypisaną pewną liczbową ocenę (pole RATE).

Ocena ta ma odzwierciedlać wagę jaką przypisuje inżynier wiedzy akcji reguły. Im wyżej oceniane są zmiany wprowadzane w tablicy przez akcję tym wyższą wartość ma ta ocena (dopuszczalne są wartości z przedziału od 0 do 100).

## MODUŁY STERUJĄCE

W tablicowym systemie eksperckim rozwiązanie budowane jest krok po kroku na skutek ciągu wywołań źródeł wiedzy. Współpraca źródeł wiedzy, reprezentujących niezależnych ekspertów opiera się na założeniu, że informacja niezbędna do uzyskania rozwiązania znajdzie się w najdogodniejszym miejscu i czasie w tablicy, dostarczona przez kogoś spośród nich [4]. Żadne bowiem źródło wiedzy nie jest w stanie samo rozwiązać problemu postawionego przed systemem. Aby jednak tak się stało, potrzebne są moduły sterujące, które określą następne źródło wiedzy, biorące udział w przetwarzaniu.

Przebieg sterowania systemem tablicowym można opisać następującym ciągiem decyzji podejmowanych przez poszczególne moduły sterujące:

- a) monitor zdarzeń wykrywa zmiany dokonane w tablicy przez źródło wiedzy i spośród nich wybiera tylko te, które zaliczone zostały przez projektanta aplikacji do zdarzeń,
- b) menadżer agendy wybiera spośród dotychczas nie analizowanych zdarzeń jedno, które posłuży za kontekst wywołania następnego źródła wiedzy,
- c) moduł szeregujący spośród wielu źródeł wiedzy reagujących na wybrane powyżej zdarzenie dokonuje wyboru jednego źródła wiedzy,
- d) menadżer źródeł wybiera wiązania dla zmiennych obiektowych zdefiniowanych w bazie wiedzy wybranego źródła, a następnie uaktywnia źródło,
- e) wywołane źródło wiedzy modyfikuje tablicę i cały proces się powtarza.

System tablicowy kończy pracę kiedy pojawi się predefiniowane zdarzenie STOP lub brak źródeł gotowych do dalszej współpracy. Drugi przypadek ma miejsce wtedy, gdy menadżer agendy nie ma do wyboru żadnych zdarzeń i/lub moduł szeregujący nie ma do dyspozycji żadnego źródła wiedzy, które zareagowało na wybrane zdarzenie.

## MONITOR ZDARZEŃ I MENADŻER AGENDY

Wyboru następnego zdarzenia, które posłuży jako bodziec do wywołania odpowiednich źródeł wiedzy dokonuje się w dwóch etapach. W pierwszym etapie *monitor zdarzeń* na podstawie definicji zdarzeń podanej przez projektanta aplikacji odrzuca wszystkie te zmiany w tablicy, które do zdarzeń nie należą. W drugim etapie *menadżer agendy* spośród tak okrojonego zbioru zmian wybiera tylko jedno zdarzenie do dalszego przetwarzania zgodnie z algorytmem  $\Delta$ -MIN Carbonell'a opisanym w [1]. Funkcjonowanie menadżera agendy w swej istocie nie odbiega od sposobu działania charakteryzującego systemy sterowane tzw. agendą (ang. agenda-driven systems). Najważniejszą zaletą algorytmu  $\Delta$ -MIN jest to, iż zapewnia w sytuacji nawrotu powrót do zmiany ocenionej najwyższej spośród **wszystkich** dotychczas wygenerowanych zmian, a nie tylko zmian, których autorem było ostatnio uaktywnione źródło.

## MODUŁ SZEREGUJĄCY

Wyboru jednego źródła spośród wszystkich tych źródeł, które zareagowały na zdarzenie wybrane przez menadżera agendy dokonuje moduł szeregujący, korzystając w tym celu z podstawowych informacji o źródle zawartych w strukturze zwanej *rekordem aktywacji źródła* wiedzy (ang. KSAR - Knowledge Source Activation Record).



KSAR	
NAME	ksar_21
KS_NAME	constraints_checker
TRIGGERING_CYCLE	63
TRIGGERING_EVENT	event_15
METHOD	forward
COMPLEXITY	27
IMPORTANCE	0.8
PRIORITY	0.5

Rys. 6. Przykładowy rekord aktywacji dla źródła z rys. 4

Część informacji umieszczona w rekordzie aktywacji (rys. 6) pochodzi z definicji źródła wiedzy. Pola: KS\_NAME, METHOD, COMPLEXITY i IMPORTANCE są danymi zaczerpniętymi bezpośrednio z deskryptora źródła. Pozostałe wartości określone są w trakcie pracy systemu. Należą do nich:

- nazwa rekordu aktywacji (pole NAME) - unikalny identyfikator rekordu aktywacji (do użytku wewnątrz systemu),
- cykl aktywacji (pole TRIGGERING\_CYCLE) - stan licznika głównej pętli sterowania z chwili, kiedy wystąpiło zdarzenie uaktywniające,
- zdarzenie uaktywniające (pole TRIGGERING\_EVENT) - wskaźnik do zdarzenia, na które zareagowało źródło (inaczej: zdarzenia, które wybrał menadżer agendy),
- priorytet (pole PRIORITY) - miara, według której dokonuje się wyboru najlepszego źródła spośród uaktywnionych.

Wśród opisanych danych najważniejszą funkcję pełni *priorytet*, gdyż spośród wszystkich źródeł, które zareagowały na zdarzenie wybierane jest to, dla którego jego wartość jest najwyższa. Wartość priorytetu obliczana jest w trakcie tworzenia rekordu aktywacji jako suma ważona trzech wielkości: złożoności źródła wiedzy, znaczenia źródła i oceny zdarzenia, na które zareagowało źródło. Odpowiednie współczynniki potrzebne do obliczenia sumy ważonej (KS\_IMPORTANCE, KS\_COMPLEXITY, TRIGGER\_RATE) są definiowane przez projektanta aplikacji.

#### MENADŻER ŹRÓDEŁ

W dowolnej chwili pracy systemu eksperckiego w tablicy może znajdować się wiele obiektów tej samej klasy. Zanim jednak zostanie użyta wiedza przedmiotowa wyrażona w postaci reguł należy zdecydować, których obiektów będą dotyczyć wnoszone zmiany. W tym celu musimy dokonać skojarzenia zmiennych zawartych w bazie z konkretnymi obiektami w tablicy. Zadaniem tym zajmuje się menadżer źródeł.

Najpierw dla każdej zmiennej z deklaracji źródła (pole VARIABLE rys. 5) wyszukiwane są wszystkie możliwe wiązania (skojarzenia) i tworzone jest drzewo alternatywnych skojarzeń. Potem dla każdej zmiennej wybierane jest jedno wiązanie - zbiór skojarzeń wszystkich zmiennych reprezentowany jest w drzewie jako ścieżka. Po wyborze wiązań uruchamiany jest proces wnioskowania źródła. Jeśli zachodzi potrzeba zmiany wiązania drzewo skojarzeń przeglądane jest zgodnie z typową strategią przeszukiwania w głąb.

#### PODSUMOWANIE

W pracy scharakteryzowano podstawowe elementy architektury uczestniczące w sterowaniu tablicowym systemem eksperckim. Skoncentrowano się na opisie sposobu funkcjonowania



modułów sterujących. Zaproponowane rozwiązania umożliwiają znaczną modyfikację zachowania systemu w trakcie pracy. Deklaracje zdarzeń, ich wystąpienia na liście warunków aktywacji źródeł, oceny przypisywane regułom (a w konsekwencji zdarzeniom), współczynniki potrzebne do obliczenia priorytetu rekordu aktywacji, listy wiązań lokalnych baz wiedzy - wszystkie te elementy dają dużą swobodę w projektowaniu systemu tablicowego dla określonego zastosowania.

## LITERATURA

- [1] CARBONELL J.G., DELTA-MIN: A Search-Control Method for Information-gathering Problems. w *Proc. AAAI 1*, 1980.
- [2] HAYES-ROTH B., A Blackboard Architecture for Control, *Artificial Intelligence* **26** (1985) pp.251-321.
- [3] HUMPERT B., DE KORVIN A., Designing and Implementing Blackboard Architectures, *Expert Systems for Information Management* vol. 3, No 3, 1990, pp.165-194.
- [4] NII H.P., Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures, *The AI Magazine*, Summer 1986, pp.38-53.
- [5] NII H.P., Blackboard Systems Blackboard Application Systems, Blackboard Systems from Knowledge Engineering Perspective, *The AI Magazine*, August 1986, pp.82-106.

**ISBN 83-85847-85-5**

---

**W celu uzyskania bliższych informacji i zakupu dodatkowych egzemplarzy  
prosimy o kontakt  
z Instytutem Badań Systemowych PAN  
ul. Newelska 6, 01-447 Warszawa  
tel. 36-19-01 w. 241 e-mail: kotuszew@ibspan.waw.pl**