

Paweł SITEK, Jarosław WIKAREK
Politechnika Świętokrzyska, Katedra Systemów Sterowania i Zarządzania

ZASTOSOWANIE ŚRODOWISK DEKLARATYWNYCH DO WSPOMAGANIA ZARZĄDZANIA PROJEKTAMI

Streszczenie. Zarządzanie projektem należy do ważnej grupy problemów decyzyjnych, które charakteryzują się wieloma ograniczeniami. Tradycyjne podejścia do rozwiązywania tych problemów bazujące na programowaniu matematycznym są często nieefektywne ze względu na trudność modelowania ograniczeń oraz złożoność obliczeniową. W odróżnieniu od podejścia tradycyjnego zastosowanie deklaratywnego środowiska programowania w logice z ograniczeniami (Constraint Logic Programming) do modelowania i rozwiązywania problemów decyzyjnych, w których występują ograniczenia, jest naturalne. W pracy zaproponowano wykorzystanie oraz wzajemną integrację środowisk deklaratywnych (SQL, CLP) do przedstawienia ogólnej koncepcji systemu wspomaganie decyzji. Na podstawie tej koncepcji dokonano implementacji systemu wspomaganie decyzji zarządzania projektem.

Słowa kluczowe: wspomaganie decyzji, CSP (Constraint Satisfaction Problem), CLP (Constraint Logic Programming), relacyjne bazy danych

A DECLARATIVE APPROACH TO BUILDING DECISION SUPPORT SYSTEM FOR PROJECT MANAGEMENT

Summary. Project management is usually characterized by many types of constraints, which make this unstructured and difficult to solve. Traditional mathematical programming approaches are deficient because their representation of constraints is artificial (using 0-1 variables). Unlike traditional approaches, constraint logic programming (CLP) provides for a natural representation of heterogeneous constraints. In this paper we propose a declarative framework for decision support system (DSS) for project management implemented by CLP and relational SQL database.

Keywords: decision support, CSP, CLP, relational database

1. Wstęp

Zarządzanie projektem jest złożonym procesem planowania, organizowania oraz zarządzania zadaniami i zasobami, w wyniku którego osiągany jest założony cel w ramach istniejących ograniczeń (czasowych, zasobowych, kosztowych itp.). Problem zarządzania projektem zwykle zaczyna się od budowy planu projektu, tzn. od określenia listy zadań, czasu ich trwania oraz zależności pomiędzy nimi. W kolejnym kroku należy uwzględnić zasoby niezbędne do wykonania zadań. Na końcu procesu należy znaleźć odpowiedni przydział zasobów do zadań oraz kolejność ich wykonania (jeśli nie jest jednoznacznie narzucona na wstępie).

Tak sformułowany problem jest niezwykle interesujący w kontekście zagadnień występujących w praktyce. W wielu dziedzinach można wprost znaleźć problemy zarządzania projektem. Są to zarówno projekty obiektowe np. projektowanie i wznoszenie obiektów budowlanych, projektowanie i wykonywanie unikatowych produktów technicznych, produkcja filmów, instalacja specjalistycznego wyposażenia, wprowadzenie nowego produktu oraz projekty procesowe, np. wprowadzenie nowego systemu zarządzania, projektowanie i wdrażanie systemów informatycznych, realizacja nowej strategii sprzedaży itp.

2. Koncepcja wspomaganie decyzji zarządzania projektami

Biorąc pod uwagę różnorodność problematyki zarządzania projektami oraz konieczność uwzględnienia dodatkowych specyficznych ograniczeń a także złożoność obliczeniową modelowanych problemów zaproponowano wykorzystanie środowisk deklaratywnych do opracowania podstawowych struktur systemu wspomaganie zarządzania projektami. Środowiska deklaratywne są oparte na paradygmacie programowania deklaratywnego. W środowiskach tych program składa się najczęściej nie z podstawień, sterowania przepływem (pętle, instrukcje warunkowe, wyboru itp.), lecz z deklaracji. W odróżnieniu od języków imperatywnych, gdzie należy znaleźć sposób rozwiązania problemu (algorytm), czyli odpowiedzieć na pytanie „*jak rozwiązać problem*”, w językach deklaratywnych najczęściej odpowiadamy na pytanie „*jaki jest problem*”.

Ogólne założenia funkcjonalne systemu wspomaganie decyzji zarządzania projektami pokazano na rys. 1. Wejściem do systemu jest zbiór zadawanych pytań. Zadawane pytania mogą być dwojakiego typu. Pierwszy typ pytań (P1) dotyczy poszukiwania rozwiązania spełniającego określone kryterium jakości, np. *Jaki jest minimalny czas realizacji projektu? Jaka jest minimalna liczba ludzi do wykonania zbioru zadań w ramach projektu?* Drugi typ pytań (P2) dotyczy istnienia rozwiązań, np. *Czy jest możliwa realizacja projektu w określonym*

czasie? Sercem systemu jest deklaratywne środowisko CLP (Constraint Logic Programming) [3], które powstało w wyniku modyfikacji programowania w logice, gdzie prostą zasadę rezolucji [2] zastąpiono problemem spełnienia ograniczeń CSP (Constraint Satisfaction Problem). W środowisku CLP opracowano następujące zbiory predykatów:

- implementujące zbiór pytań wejściowych (PW),
- transformacji danych z relacyjnej bazy danych (TD),
- automatycznej generacji modeli CLP, które wykorzystują dwa pierwsze typy predykatów (AGM).

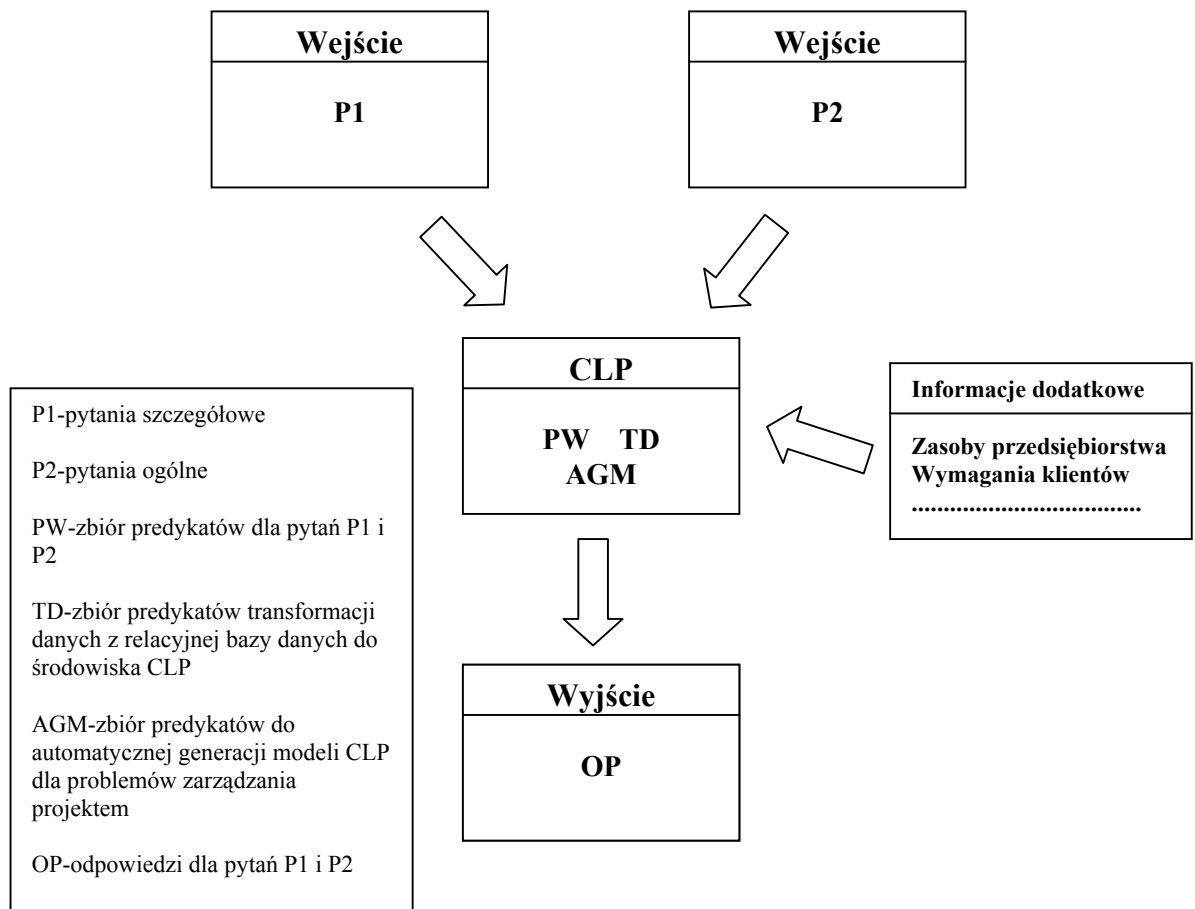
Wyjściem systemu są odpowiedzi na zadane pytania (odpowiednio sformułowane predykaty-OP) oraz wartości zmiennych, na podstawie których budowane są wykresy Gantta. W prezentowanej wersji systemu wspomaganie decyzji zarządzania projektem mogą być zadawane między innymi następujące pytania (wywoływane odpowiednie predykaty CLP):

- Jaka jest minimalna liczba pracowników potrzebna do wykonania zadań w określonym czasie? (predykat $opc_d(L,C)$).
- Jaki jest minimalny czas realizacji zadań przy określonej liczbie pracowników? (predykat $opc_g(L,C)$).
- Czy jest możliwe wykonanie zbioru zadań w określonym terminie przy określonej liczbie pracowników? (predykat $opc_s(L,C)$).

C oznacza długość uszeregowania, natomiast L to liczba pracowników.

Powyższe pytania mogą uwzględniać dowolny zbiór dodatkowych zasobów (w tym przypadku są to tylko pracownicy). Dodatkowo, w przedstawianej wersji systemu, wspomaganie decyzji została zaimplementowana dodatkowa funkcjonalność, polegająca na rozdziale obciążeń. W problemach opisanych w literaturze czas realizacji poszczególnych zadań jest na ogół stały i określony przed ich wykonaniem. W zastosowaniach praktycznych bardzo często czas wykonania poszczególnych zadań zależy w istotny sposób od ilości przydzielonych zasobów do ich wykonania. Zależności te na ogół są nieliniowe, można je przedstawić za pomocą relacji (tabeli relacyjnej bazy danych) lub funkcji. W systemie zaimplementowano możliwość zmiany czasu wykonania zadań w zależności od przydzielonej liczby pracowników. W ogólności, może to być dowolny typ zasobu lub zbioru zasobowego. Powyższa funkcjonalność nie wymaga zmiany predykatów, a jedynie odpowiednio przygotowanych danych opisujących problem, znajdujących się w relacyjnej bazie danych. Zaproponowana struktura relacyjnej bazy danych [1] oraz sposób budowy predykatów CLP powodują, że prezentowana koncepcja może być podstawą do budowy systemów wspomaganie decyzji nie tylko przy zarządzaniu projektem, ale również planowania i harmonogramowania produkcji o różnym typie organizacji produkcji (job-shop, flow-shop, project itp.). Można również uwzględniać rozdział dodatkowych zasobów, np. ludzi, narzędzi, środków finansowych itp. Uwzględnienie dodatkowych zasobów oraz ograniczeń na nie nakładanych jest niedostępne

przy wykorzystaniu np. klasycznych metod zarządzania projektem typu CPM (Critical Path Method) czy PERT (Program Evaluation and Review Technique).



Rys. 1. Koncepcja systemu wspomaganie decyzji
Fig. 1. The concept of the decision support system

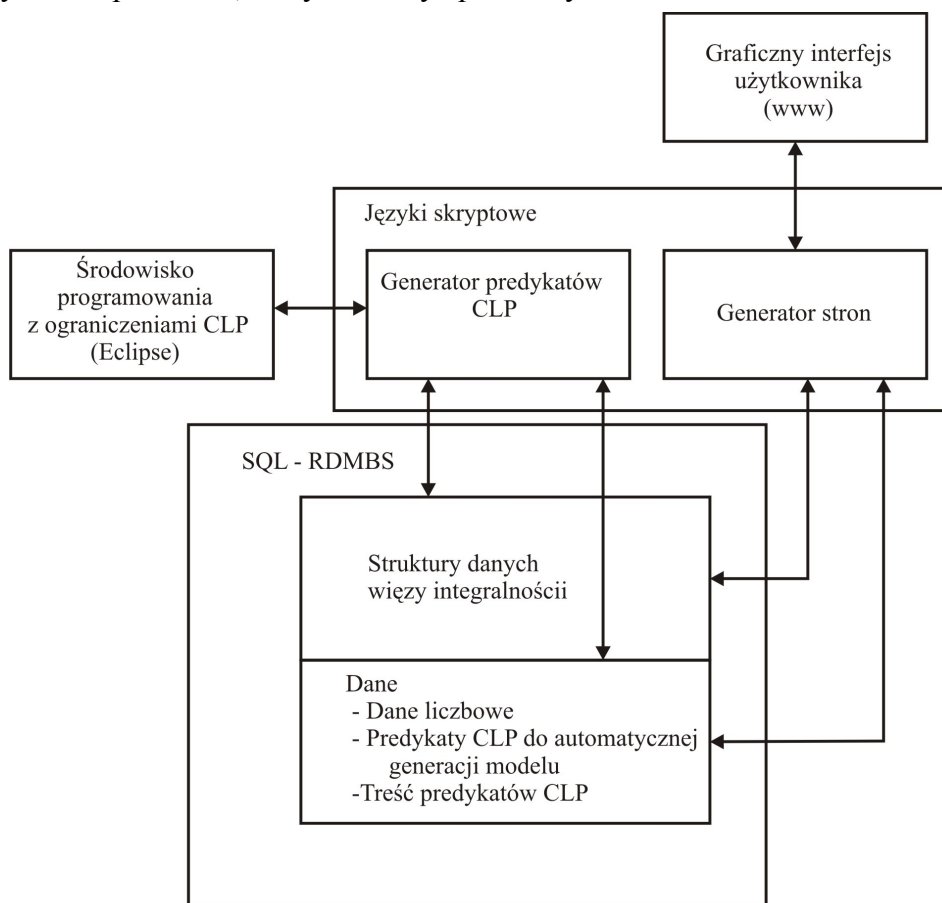
3. Implementacja podstawowych struktur systemu wspomaganie decyzji zarządzania projektem

Do implementacji systemu zastosowano środowiska deklaratywne. System wykonano w architekturze wielowarstwowej (rys. 2). W warstwie bazy danych zastosowano relacyjną bazę danych z deklaratywnym językiem SQL – dostępu do danych. W warstwie logiki przetwarzania zastosowano środowisko CLP – ECLiPSe [3, 4, 5, 6]. Interfejs użytkownika oprogramowano w języku HTML. Strony są generowane dynamicznie. Zastosowanie środowisk deklaratywnych w każdej z warstw oraz modelu wielowarstwowego cechuje wiele zalet. Do najważniejszych z nich należy zaliczyć:

- Oddzielenie modelu od danych.
- Możliwość zdalnego korzystania z systemu (przez sieć WAN, Intranet, Internet).

- Łatwą modyfikację i rozbudowę części logiki systemu przez dodanie nowych predykatów.
- Możliwość zaprojektowania schematu bazy danych, w którym mogą być zapisane dane dla różnych typów problemów harmonogramowania produkcji.
- Łatwą modyfikację modelu CLP przez dodanie nowych ograniczeń (bez konieczności modyfikacji modeli już istniejących).
- Małe wymagania co do sprzętu i oprogramowania, które występują po stronie klienta systemu – wystarczy komputer z przeglądarką internetową.

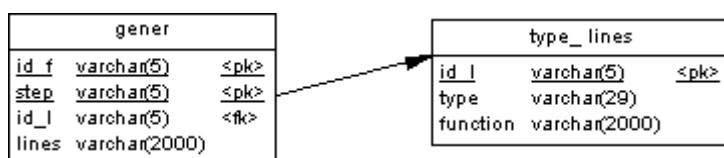
Istotną nowością prezentowanej implementacji jest integracja dwóch środowisk deklaratywnych RDBMS (Relational Database Management System) oraz CLP. Integracja ta polega między innymi na możliwości automatycznej generacji skryptów CLP na podstawie danych zawartych w tabelach relacyjnej bazy danych [1] oraz łączeniu ich razem z predykatami bezpośrednio zapisanymi w bazie danych (np. predykaty dla poszczególnych typów zadawanych pytań). Tak wygenerowane skrypty zawierają oba typy predykatów CLP i stanowią kompletny model problemu, który może być przesłany do silnika CLP.



Rys. 2. Schemat implementacyjny systemu wspomaganie decyzji

Fig. 2. The implementation schema of decision support system

W celu zapewnienia automatycznej generacji modelu problemu decyzyjnego w postaci skryptu z predykatami CLP dodano do bazy danych dwie dodatkowe tabele (rys. 3) [1]. W tabeli *gener* opisano schemat modelu w postaci wierszy zawierających identyfikator modelu (*id_f*), kroku generacji (*step*) oraz identyfikatora typu linii, jaka ma być wpisana do skryptu CLP (*id_l*) wraz z jej źródłem (*lines*). Typ generowanej linii jest ustalany na podstawie wpisu w tabeli *type_lines*. W modelu (skrypcie CLP) można wyróżnić linie powstałe między innymi jako: wstawienie predykatu CLP (wiersza z tabeli *eclipse_predicates*), wstawienie danych po wykonaniu zapytania SQL, wstawienie komentarza, nagłówka itp., stąd zależność pomiędzy tabelami *gener* i *type_lines* jest typu 1:N (rys. 3). Opis struktury bazy danych przedstawiono w tabeli 1.



Rys. 3. Fragment schematu struktury bazy danych systemu wspomaganie decyzji
Fig. 3. The part of database schema for decision support system

Tabela 1

Opis fragmentu struktury bazy danych systemu wspomaganie decyzji

Tabela	Opis	Kolumny	Klucz	Opis kolumny
Type_of_lines	Typy linii modelu	id_l	P	Identyfikator typu linii
		type		Nazwa typu linii
		function		Kod do wykonania przy generacji
Gener	Opis generacji modelu CLP	id_f	P	Identyfikator projektu
		step	P	Krok
		id_l	F	Identyfikator typu linii
		line		Linia do wygenerowania
P – klucz główny, F – klucz obcy				

4. Przykład ilustracyjny zastosowania systemu

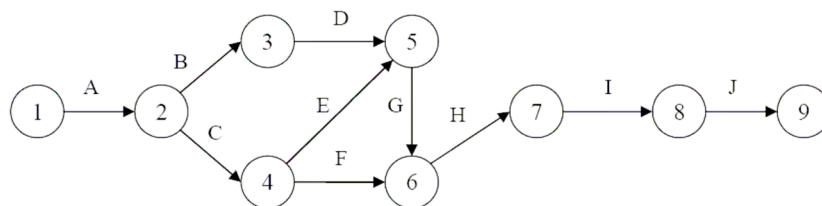
W celu zilustrowania możliwości systemu wspomaganie decyzji zarządzania projektem przedstawiono przykład liczbowy. Przykład dotyczy projektu, którego sieć działań przedstawiono na rys. 4. Podstawowe dane liczbowe dla przykładu zawarte są w tabeli 2. Przykład został rozwiązany w dwóch wersjach (*przykład_1a*, *przykład_1b*). W *przykład_1a* nie uwzględniono rozdziału zasobów w postaci przydziału dodatkowych pracowników. W *przykład_1b* już ten przydział i jego wpływ na czas trwania zadań uwzględniono (tabela 4). W obu wersjach uruchomiono predykatory *opc_g()*, *opc_d()*. Dla porównania uzyskanych wyników rozwiązano *przykład_1a* metodą CPM (ścieżki krytycznej). To porównanie dotyczy je-

dynie wyników uzyskanych przy uruchomieniu predykatu $opc_g(,)$. Uruchomienie innych predykatów wiązało się z koniecznością uwzględnienia ograniczeń na posiadane zasoby lub zależności czasu wykonanych zadań od przydzielonych zasobów, co jest niedostępne dla metody CPM. Wszystkie wyniki, tzn. minimalny czas realizacji C_{max} oraz liczbę pracowników L_{min} przedstawiono w formie tabelarycznej (tabela 3 i 5) oraz wykresów Gantta (rysunki 5, 6 i 7).

Tabela 2

Dane liczbowe *przykład 1a*

Zadanie	Poprzednik	Czas trwania zadania	Pracownicy
A	Brak	5	1
B	A	7	2
C	A	4	1
D	B	3	1
E	C	8	2
F	C	7	1
G	D E	2	1
H	F, G	5	1
I	H	6	2
J	I	4	1

Rys. 4. Kolejność wykonywania zadań dla *przykład_1a* i *przykład_1b*Fig. 4. The dependencies between the activities for *przykład_1a* i *przykład_1b*

Rozwiązanie metodą ścieżki krytycznej CPM:

$$T_1(w) = 0$$

$$T_2(w) = 0 + 5 = 5$$

$$T_3(w) = 5 + 7 = 12$$

$$T_4(w) = 5 + 4 = 9$$

$$T_5(w) = \max\{12 + 3, 9 + 8\} = 17$$

$$T_6(w) = \max\{17 + 2, 9 + 7\} = 19$$

$$T_7(w) = 19 + 5 = 24$$

$$T_8(w) = 24 + 6 = 30$$

$$T_9(w) = 30 + 4 = 34$$

$$T_9(k) = 34$$

$$T_8(k) = 34 - 4 = 30$$

$$T_7(k) = 30 - 6 = 24$$

$$T_6(k) = 24 - 5 = 19$$

$$T_5(k) = 19 - 2 = 17$$

$$T_4(k) = \min\{17 - 8, 19 - 7\} = 9$$

$$T_3(k) = 17 - 3 = 14$$

$$T_2(k) = \min\{14 - 7, 9 - 4\} = 5$$

$$T_1(k) = 5 - 5 = 0$$

Tabela 3

Wyniki dla przykład 1a

Zadanie	Metoda CPM	$opc_g(,)$ Czas startu	$opc_d(,34)$ Czas startu	$opc_g(3,)$ Czas startu
	$C_{max}=34$	$C_{max}=34$	$L_{min}=34$	$C_{max}=37$
A	0	0	0	0
B	5	5	5	5
C	5	5	5	5
D	14	12	12	16
E	9	9	9	12
F	9	9	12	9
G	17	17	17	20
H	19	19	19	22
I	24	24	24	27
J	30	30	30	33

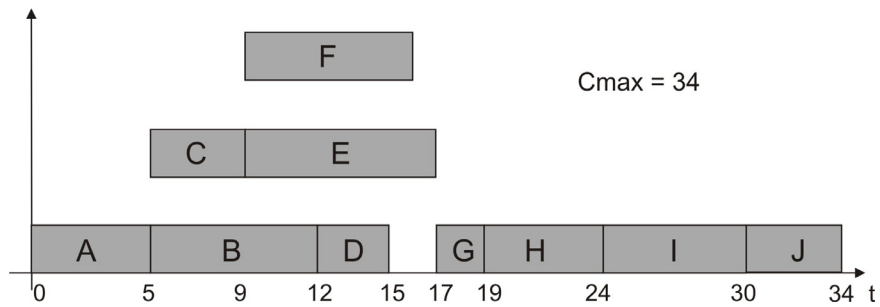
Rys. 5. Wykres Gantta dla przykład 1a (metoda ścieżki i predykat $opc_g(,)$)Fig. 5. The Gantt chart for przykład 1a (CPM and predicate $opc_g(,)$)

Tabela 4

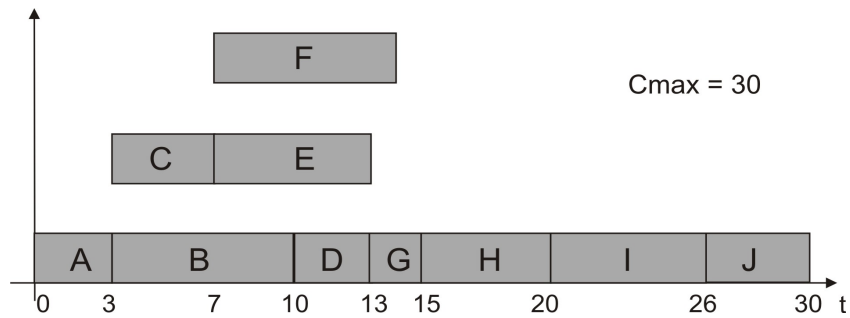
Dodatkowe dane liczbowe przykład 1b

Zadania	Dodatkowi pracownicy			
	Czas	Pracownicy	Czas	Pracownicy
A	5	1	3	2
B	7	2	5	4
C	4	1	---	-----
D	3	1	---	-----
E	8	2	6	3
F	7	1	5	3
G	2	1	---	-----
H	5	1	---	-----
I	6	2	---	-----
J	4	1	---	-----

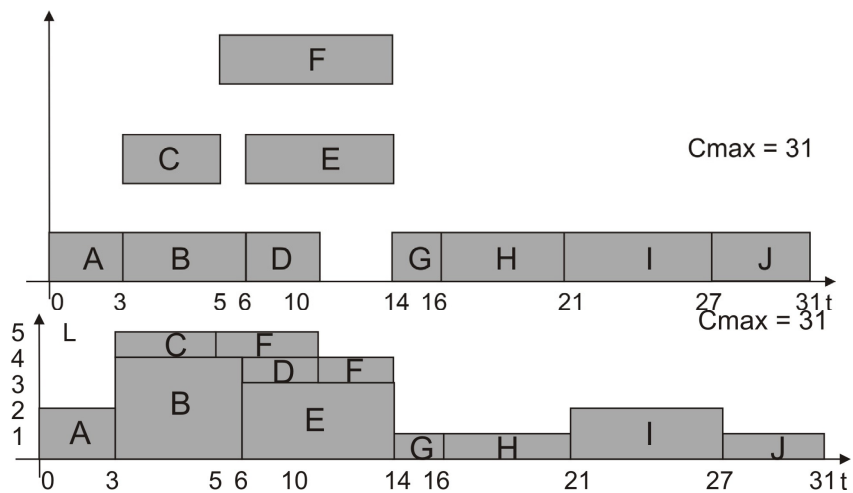
Tabela 5

Wyniki przykład 1b

Zadanie	$opc_g(_,_) C_{max}=30,$ $opc_d(_,30) L_{min}=6$			$opc_g(5,_) C_{max}=31$		
	Długość	Pracownicy	Start	Długość	Pracownicy	Start
A	3	2	0	3	2	0
B	7	2	3	5	4	3
C	4	1	3	4	1	3
D	3	1	10	3	1	8
E	6	3	7	6	3	8
F	7	1	7	7	1	7
G	2	1	13	2	1	14
H	5	1	15	5	1	16
I	6	2	20	6	2	21
J	4	1	26	4	1	27



Rys. 6. Wykres Gantta dla przykład_1b (predykat $opc_g(_,_)$)
 Fig. 6. The Gantt chart for przykład_1b (predicate $opc_g(_,_)$)



Rys. 7. Wykres Gantta dla przykład_1b (predykat $opc_g(5,_)$)
 Fig. 7. The Gantt chart for przykład_1b (predicate $opc_g(5,_)$)

5. Podsumowanie

Zaproponowana architektura systemu wykorzystująca środowiska deklaratywne jest interesującą propozycją w kontekście budowy systemów wspomaganie decyzji. System taki można bowiem stosować do wielu problemów decyzyjnych planowania i harmonogramowania projektów i produkcji niezależnie od formy ich organizacji. Można również przy podejmowaniu decyzji uwzględniać specyficzne ograniczenia rozważanego problemu (przez dodanie nowych predykatów), dodatkowe/zewnętrzne zasoby, rozdział obciążeń, czasową niedostępność zasobów itp. W odróżnieniu od klasycznych rozwiązań, gdzie do każdego z proponowanych zagadnień należy zbudować model problemu i znaleźć metodę jego rozwiązania, w programowaniu deklaratywnym samo sformułowanie problemu wystarcza do jego rozwiązania. Istotną wartością prezentowanego rozwiązania jest zaproponowanie wzajemnej integracji środowisk deklaratywnych, a szczególnie relacyjnej bazy danych i środowiska CLP oraz sposobu automatycznej generacji modeli problemów decyzyjnych w postaci predykatów na podstawie danych znajdujących się w bazie danych. Zastosowanie w warstwie logiki przetwarzania deklaratywnego środowiska CLP–ECLiPSe zapewnia dużą efektywność znajdowania rozwiązań poszczególnych problemów decyzyjnych (szczególnie rozwiązań dopuszczalnych) przez wykorzystanie metod i technik programowania z ograniczeniami, m.in. propagacji ograniczeń, backtrackingu itp.

BIBLIOGRAFIA

1. Sitek P., Wikarek J.: Zastosowanie środowisk deklaratywnych do budowy struktur systemu wspomaganie decyzji harmonogramowania produkcji. BDAS 2008. WKiŁ, s. 399÷409.
2. Ben-Ari Mordechai: Logika matematyczna w informatyce, WNT, 2004.
3. Liao S.Y., Wang H.Q., Liao L.J.: An extended formalism to constraint logic programming for decision analysis, Knowledge-based Systems 15, 2002, s. 189÷202.
4. Lee H.G., Lee G. Yu.: Constraint logic programming for qualitative and quantitative constraint satisfaction problems, Decision Support Systems 16 (1), 1996, s. 67÷83.
5. www.eclipse-clp.org.
6. Apt K., Wallace M.: Constraint Logic Programming using ECLiPSe, Cambridge 2007.

Recenzent: Dr inż. Rafał Augustyn

Wpłynęło do Redakcji 4 lutego 2009 r.

Abstract

Project management is usually characterized by many types of constraints, which make this unstructured and difficult to solve. Traditional mathematical programming approaches are deficient because their representation of constraints is artificial (using 0-1 variables). Unlike traditional approaches, constraint logic programming (CLP) provides for a natural representation of heterogeneous constraints. In CLP we state the problem requirements by constraints; we do not need to specify how to meet these requirements. In this paper we propose a declarative framework for decision support system (DSS) for project management implemented by CLP and relational SQL database. We illustrate this concept by the implementation of a DSS for project management with external resources and constraints for that resources.

Adresy

Paweł SITEK: Politechnika Świętokrzyska, Katedra Systemów Sterowania i Zarządzania, Al. 1000-PP 7, 25-314 Kielce, Polska, sitek@tu.kielce.pl.

Jarosław WIKAREK: Politechnika Świętokrzyska, Katedra Systemów Sterowania i Zarządzania, Al. 1000-PP 7, 25-314 Kielce, Polska, wikarek@tu.kielce.pl.