



POLSKA AKADEMIA NAUK
Instytut Badań Systemowych

Edward Michalewski

**PODSTAWY METODY
ANALIZY DIAGNOSTYCZNEJ
I PROJEKTOWANIA SYSTEMÓW
ZARZĄDZANIA (METODA DIANA)**

Publikację opiniowali do druku:

Prof. dr hab. inż. Ludosław Drelichowski

Prof. dr hab. inż. Piotr Sienkiewicz

Copyright © by Instytut Badań Systemowych PAN

Warszawa 2004

ISBN 83-85847-87-1

ISSN 0208-8029

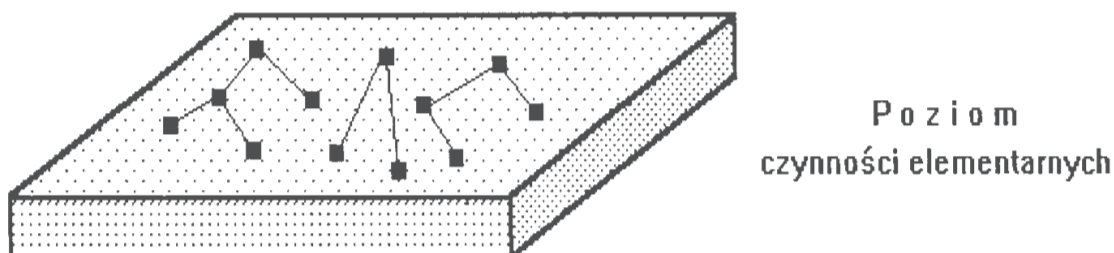
Edward Michalewski

**PODSTAWY METODY ANALIZY
DIAGNOSTYCZNEJ
I PROJEKTOWANIA SYSTEMÓW
ZARZĄDZANIA (METODA DIANA)**

III. MODEL SYSTEMU INFORMACYJNEGO ZARZĄDZANIA

III.1. Jednopoziomowy statyczny model systemu informacyjnego zarządzania

Już w pierwszej wersji metody DIANA przyjęto założenie, że model systemu informacyjnego zarządzania badanego obiektu stanowi sieć powiązań informacyjnych [76]. W sieci tej węzłami są zadania, składające się z podzadań, te zaś z elementarnych czynności (operacji) wykonywanych przez personel badanego obiektu, zaś łukami nawzajem przekazywane informacje, będące wynikami realizacji tych czynności (rys. 7.):



Rys. 7. Pierwotny model systemu informacyjnego zarządzania

Sieć ta posiada więc własną hierarchię: zadania - podzadania - operacje elementarne. Zarówno węzły jak i łuki opisywane są szeregiem parametrów (pracochłonność, okresowość, terminy rozpoczęcia i zakończenia, rodzaj realizowanej funkcji itd.), wykorzystywanych w komputerowej diagnozie do wykrywania istniejących nieprawidłowości. Na trafność takiego podejścia wskazywała wykrywalność w trakcie diagnozy tzw. ślepych uliczek informacyjnych (brak rzeczywistego odbiorcy), stanowiących przeciętnie 30% ogółu czynności w zdecydowanej większości z ponad 100 przebadanych obiektów (od małych przedsiębiorstw poczynając, na całej branży kończąc). Pierwsza wersja pakietu DIANA potrafiła w wykryć tylko dwa objawy: właśnie ten oraz wynikający z niego objaw – brak jakiegokolwiek odbiorcy. Stopniowo w następnych wersjach możliwości te zostały znacznie rozszerzone. Sieć powiązań informacyjnych była testowana przez szereg algorytmów, mających za zadanie wykrycie również innych nieprawidłowości (jak np. dublowanie czynności, wąskie gardła, brak synchronizacji, dysfunkcjonalność itd.). Było to, inaczej

mówiąc, badanie wieloaspektowe, które zakładało, że sieć powiązań informacyjnych odzwierciedla wszelkie nieprawidłowości w badanym obiekcie. Identyfikacja niedomagania implikowała usprawniające zmiany na modelu. Ten iteracyjny proces (na modelu) prowadzony był do momentu uzyskania kompromisowego rozwiązania i dopiero po tym wdrażany [92].

Przy opisie matematycznym tego modelu, już w pierwszej wersji metody DIANA, przyjęto założenie, że modelem badanego obiektu (pojęcia "systemowy" jeszcze wówczas nie używano) powinna być sieć powiązań informacyjnych. Formalnie miała ona postać [76]:

$$S = \langle K, Z, P, O, R \rangle$$

gdzie: K,Z,P,O - zbiory: komórek, zadań, podzadań i operacji;
R - relacje na tych zbiorach. przy czym:

$$S = \bigcup_{i=1}^k K, \bigcap_{i=1}^k K = \emptyset$$

$$K_i = \bigcup_{j=1}^l Z, \bigcap_{j=1}^l Z = \emptyset$$

$$Z_j = \bigcup_{r=1}^m P, \bigcap_{r=1}^m P = \emptyset$$

$$P_r = \bigcup_{s=1}^n O, \bigcap_{s=1}^n O = \emptyset$$

Z powyższego widać, że w sieci tej węzłami na 1-szym poziomie są elementy zbioru O, zaś łukami nawzajem przekazywane informacje na tym poziomie. Na poziomie 2-gim elementy zbioru P powiązane są łukami wynikającymi z relacji między elementami (łuki) na poziomie 1-szym itd. Poziom najwyższy zawiera elementy zbioru K powiązane łukami wynikającymi z relacji między elementami zbioru Z. Każdy z wyższych poziomów posiada własną strukturę hierarchiczną, powiązaną ze strukturą poziomu niższego. Poziom 1-szy ma strukturę jednorodną, zaś jego elementy opisane są szeregiem parametrów, które mają swoje odpowiedniki na wyższych poziomach w postaci odpowiednio wyliczanych agregatów.

Opis formalny tego modelu zaczniemy od jego najniższego poziomu – czynności elementarnych (operacji).

Dla ustalenia uwagi zakładamy, że badamy konkretny obiekt - system zarządzania. W związku z tym zgodnie z metodą, przedstawioną w rozdziale pierwszym, system (S) został podzielony na komórki (K), te z kolei na zadania (Z), zadania na podzadania (P), a te ostatnie na operacje (O). W tym przypadku węzłami pełnej sieci będą najdrobniejsze elementy systemu, czyli operacje. Łukami wchodzącymi będą informacje dostarczane przez inne operacje (I_D), zaś łukami wychodzącymi będą wyniki wykonania operacji (I_O) po przekazaniu ich odpowiednim odbiorcom (I_{OW}).

Dla każdej operacji, umieszczonej w teaurusie (kompletna lista czynności badanego obiektu), możemy napisać następującą zależność:

$$\bigwedge_{i=1}^k K(i) \left(\bigwedge_{j=1}^l Z(i, j) \left(\bigwedge_{r=1}^m P(i, j, r) \left(\bigwedge_{s=1}^n O(i, j, r, s) \right) \right) \right) \quad (79)$$

$$\bigvee_{i,j,r,s=1}^{k,l,m,m} \Theta(i, j, r, s) \Rightarrow I_O(i, j, r, s)$$

gdzie:

- Θ - identyfikator algorytmu wykonawczego tej operacji
- I_O - identyfikator wyniku wykonania tej operacji

Identyfikator algorytmu jest równoważny identyfikatorowi operacji, zaś identyfikator wyniku składa się poza tym z identyfikatorów podzadania, zadania, komórki i obiektu. Dlatego dla dowolnej s -tej operacji, r -tego podzadania, j -tego zadania, i -tej komórki obiektu S możemy napisać:

$$\Theta_s \Rightarrow O_s \quad (80)$$

$$I_O(i, j, r, s) \Leftrightarrow |S, K_i, Z_j, P_r, O_s|$$

Powyższa zależność wskazuje na skład identyfikatora (kodu) s - tej operacji, r - tego podzadania, j - tego zadania, i - tej komórki systemu o kodzie S .

Warunkiem zaistnienia I_O jest realizacja Θ_s , która z kolei jest możliwa w przypadku dostarczenia wszystkich informacji niezbędnych do jego wykonania. Te właśnie informacje oznaczamy przez identyfikator I_D (odpowiada niewidocznej części kodowej rekordu „Dostawca”). I_O przekazywane są do odbiorców oznaczonych przez identyfikatory K_w (odpowiada niewidocznej części kodowej rekordu „Odbiorca”)

Możemy teraz dla dowolnej operacji napisać:

$$\bigwedge_s \Theta_s \bigvee_D I_D \Rightarrow \bigvee_O I_O \bigwedge_w K_w \quad (81)$$

(Dla większej przejrzystości zapisu wszędzie, gdzie jest to możliwe będziemy pomijać zakresy indeksów).

Ponieważ

$$\bigwedge_D I_D \bigvee_{i=1}^k K(i) \Rightarrow \bigvee_O (I_{O_D} \vee I_{D_z}) \quad (82)$$

gdzie I_{D_z} - wynikowe informacje uzyskane z komórek zewnętrznych, zależność (81) określa całkowite powiązanie z dostawcami i częściowe z odbiorcami, czyli opisuje stan uzyskiwany bezpośrednio z wprowadzonych danych. Komputerowe „zamykanie” powiązań możemy zapisać w ten sposób:

$$\bigwedge_w K_w \bigvee_o (I_{O_w} \vee I_{D_z}) \Rightarrow \bigvee_{DOW} 1 (I_D = I_O = I_{O_w}) \quad (83)$$

Po wykonaniu tej czynności każda operacja figuruje teraz zarówno jako wykonawca jak też jako odbiorca i dostawca, czyli uwzględniając (81), (82), (83) możemy napisać:

$$\bigwedge_o I_o \bigvee_o I_{OD} \bigvee_o I_{O_w} \Rightarrow (I_{OD} = I_o) \wedge (I_{O_w} = I_o) \quad (84)$$

W przypadku gdy pierwszy składnik iloczynu jest nieprawdziwy mamy do czynienia z tzw. ślepą uliczką informacyjną (brak rzeczywistego odbiorcy), zaś gdy nieprawdziwy jest drugi składnik oznacza to konieczność komputerowego dopisania odbiorcy (odbiorca nieformalny) – patrz Rozdział IV. Po realizacji tej czynności dla wszystkich operacji uzyskujemy pełną sieć powiązań informacyjnych. Uzupełniona zawartościami identyfikatorów operacji sieć ta stanowi tzw. model bazowy komputerowej analizy i projektowania badanego obiektu.

Dla uzyskania dendrytów – patrz Rozdział VI, korzysta się tylko z powiązań między identyfikatorami operacji w modelu bazowym. Powiązania te są odzwierciedleniem struktury logicznej badanego obiektu rzeczywistego.

Dla dowolnego punktu startowego, czyli identyfikatora dowolnej operacji, której dendryt chcemy uzyskać, możemy napisać następującą zależność:

$$\bigwedge_s O_s \bigvee_o I_{O_s} \Rightarrow \bigwedge_o I_{O_s} \bigvee_D I_{OD_s} \Rightarrow \bigwedge_D I_{OD_s} \bigvee_o I_{O_{(s-1)}} \Rightarrow \quad (85)$$

$$\Rightarrow \bigwedge_o I_{O_{(s-1)}} \bigvee_D I_{OD_{(s-1)}} \Rightarrow \dots \Rightarrow \bigwedge_o I_{O_{(s_0)}} \bigvee_D (I_{OD_{(s-1)}} = K_{(k+t)} \vee S_z)$$

Ostatni człon wskazuje na to, że wszyscy dostawcy są komórkami zewnętrznymi nieankietowanymi, lub spoza obiektu, co oznacza zakończenie identyfikacji dendrytu.

Również przy niektórych innych algorytmach korzysta się jedynie z identyfikatorów wielkości (jak np. przy określeniu dysfunkcjonalności). Bardziej szczegółowo omówiono te zagadnienia w Rozdziale IV.

Powyższy model posiadając wiele zalet (między innymi łatwość realizacji, możliwość stosowania standardowych procedur itd.) ma jednak poważną wadę - nie uwzględnia dynamiki procesów zachodzących w modelu bazowym. Daje on obraz statyczny wszystkich powiązań tak, jak gdyby mogły one zajść jednocześnie w jednym momencie czasu.

W celu usunięcia tej wady opracowano model dynamiczny badanego obiektu. Rozpatrzmy go kolejno dla przypadku jednej operacji, jednej komórki i dla całego obiektu.

Parametryczną sieć powiązań informacyjnych badanego obiektu można sformułować zgodnie z algebraiczną teorią automatów skończonych [2] w następujący sposób: niech dowolna S -ta operacja będzie automatem skończonym, który może posiadać dwa stany A :

$$A = \begin{cases} A_0^s = 0 & - \text{przed realizacją operacji} \\ A_1^s = 1 & - \text{po realizacji operacji} \end{cases} \quad (86)$$

Niezbędne dla realizacji operacji wielkości wejściowe będą stanowiły alfabet wejściowy automatu

$$I_D^s = \{ I_{D1}^s, I_{D2}^s, \dots, I_{Df}^s, \dots, I_{Dp}^s \} \quad (87)$$

Alfabetem wyjściowym jest wynik realizacji operacji I_{Os} .

Zakładamy, że każda wielkość wejściowa jest zapamiętywana aż do momentu zmiany stanu automatu. W przeciwnym przypadku zgodnie ze strukturą klasycznego automatu skończonego należałoby ustalić tyle stanów ile możliwych jest kombinacji wielkości wejściowych.

Funkcje charakterystyczne automatu możemy określić następująco [20]:

Funkcja przejścia

$$A^s(v+1) = \delta_0 \left(I_{D_f}^s(v), A^s(v) \right) \quad (88)$$

gdzie: $I_{D_f}^s \in I_D^s$, $(f = 1, \dots, p)$, $A^s \in A$

$$\delta_0 = \begin{cases} A_0^s, & \text{gdy } \bigvee_{f=1}^p I_{D_f}^s(v) = 0 \\ A_1^s, & \text{gdy } \bigwedge_{f=1}^p I_{D_f}^s(v) = 1 \end{cases} \quad (89)$$

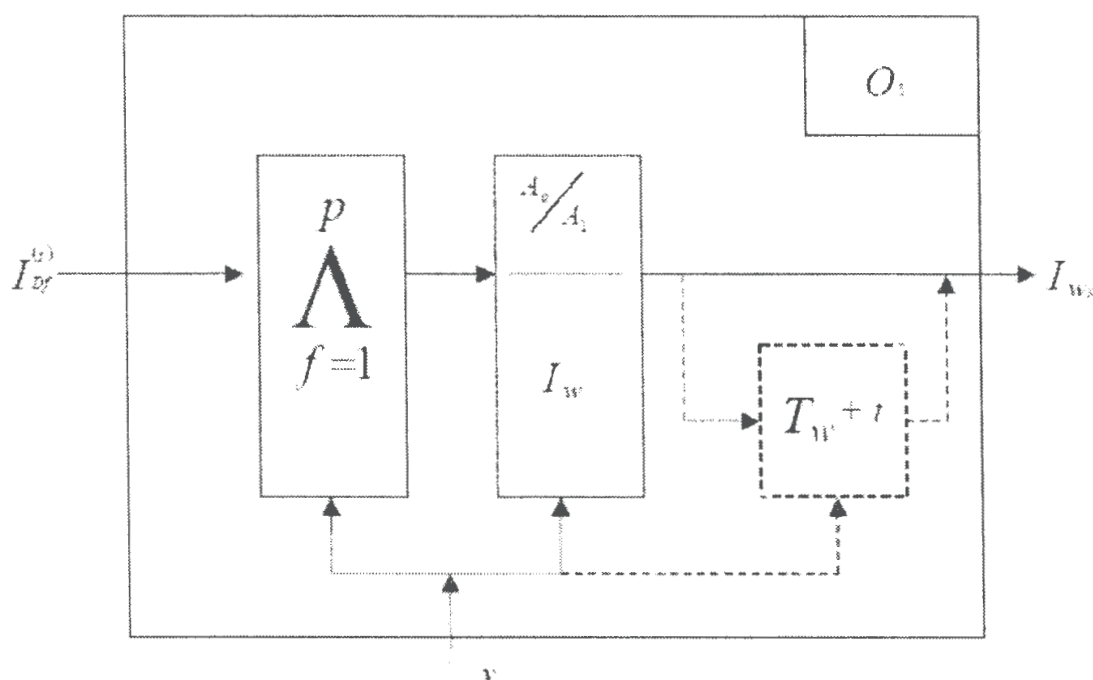
Funkcja wyjścia:

$$I_0(v) = \lambda_0 \left(I_D^s(v), A^s(v) \right) \quad (90)$$

gdzie

$$\lambda_0 = \begin{cases} 0, & \text{gdy } A^s(v) = A_0^s \rightarrow \bigvee_{f=1}^p I_{D_f}^s(v) = 0 \\ I_{0s}, & \text{gdy } A^s(v) = A_1^s \rightarrow \bigwedge_{f=1}^p I_{D_f}^s(v) = 1 \end{cases} \quad (91)$$

Schematycznie taki automat przedstawiono na Rys. 8.



Rys. 8. Automat – wersja podstawowa

Zwróćmy uwagę, że realizacja takiego automatu jest możliwa jedynie po uzyskaniu kompletnej informacji o dostawcach, a więc w przypadku pełnej (i zweryfikowanej) sieci powiązań informacyjnych. W jakim stopniu jest to konieczne przekonamy się nieco później.

Możemy teraz przeprowadzić kolejno agregację operacji na poziomie powiązań, a następnie zadań i wreszcie komórek. Istnieją w tym przypadku dwie możliwości:

- zachowanie automatu w podanej wyżej postaci elementarnej i wówczas komórka będzie posiadała tyle takich automatów ile ma do wykonania operacji, bądź też
- utworzenie jednego quasi-automatu dla komórki, który będzie posiadał tyle wejść i wyjść ile jest operacji.

Ze względu na możliwości i efektywność analizy badanego obiektu ciekawszy wydaje się drugi wariant. W tym przypadku alfabet wejściowy będzie się składał ze zbioru identyfikatorów wszystkich operacji dostawczych dla danej komórki, czyli:

$$I_D = \left\{ I_{D1}^{(1,1,1)}, \dots, I_{Df}^{(j,r,s)}, \dots, I_{Dp}^{(l,m,n)} \right\} \quad (92)$$

Alfabetem wyjściowym będzie zbiór identyfikatorów wyników wszystkich operacji komórki:

$$I_O = \left\{ I_{O_1}^{(1,1)}, \dots, I_{O_s}^{(j,r)}, \dots, I_{O_n}^{(l,m)} \right\} \quad (93)$$

Zbiór stanów będzie się składał ze stanów wszystkich operacji komórki

$$A^i = \left\{ A_0^{(1,1,1)}, A_1^{(1,1,1)}, \dots, A_0^{(j,r,s)}, \dots, A_1^{(l,m,n)} \right\} \quad (94)$$

zaś częstotliwość kroku określona jest przez najmniejszą okresowość operacji dla danej komórki.

Analogicznie (88) i (89) funkcja przejścia ma postać:

$$A^{(j,r,s)}(v+1) = \delta_k \left(I_{D_f}^{(j,r,s)}(v), A^{(j,r,s)}(v) \right) \quad (95)$$

gdzie:

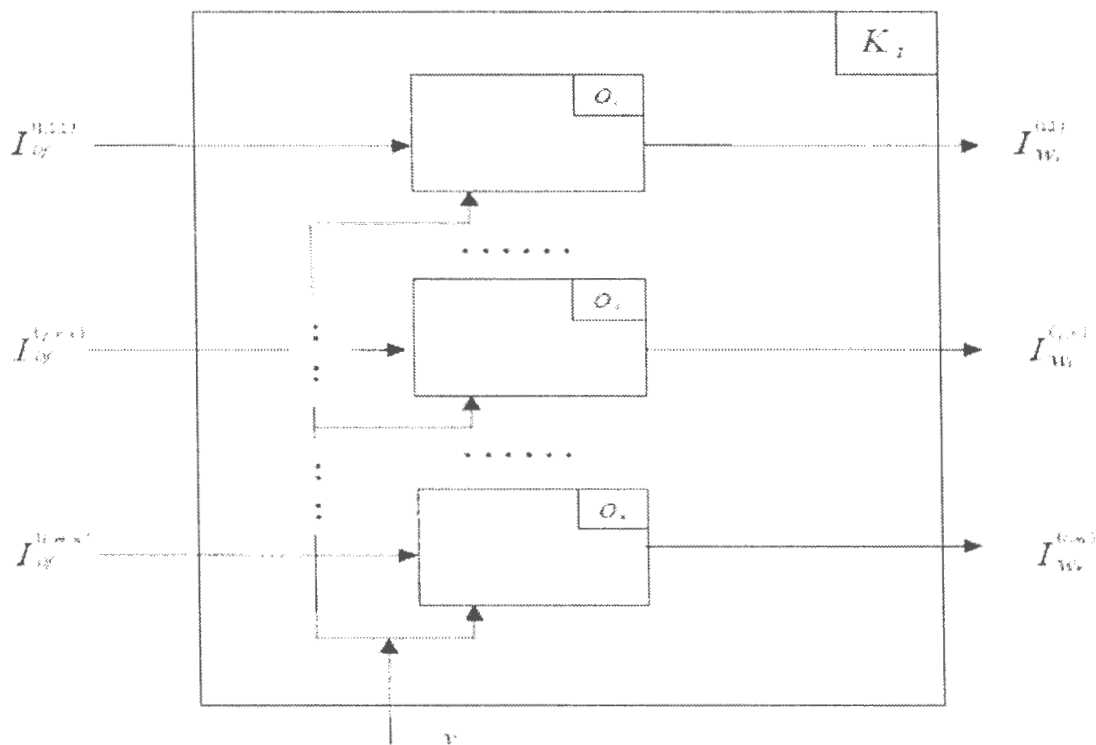
$$\delta_k = \begin{cases} A^{(j,r,s)} = A_0^{(j,r,s)} = 0, & \text{gdy } \bigvee_{j=1}^1 \bigvee_{r=1}^m \bigvee_{s=1}^n \bigvee_{f=1}^p I_{D_f}^{(j,r,s)}(v) = 0 \\ A^{(j,r,s)} = A_1^{(j,r,s)} = 1, & \text{gdy } \bigwedge_{j=1}^l \bigwedge_{r=1}^m \bigwedge_{s=1}^n \bigwedge_{f=1}^p I_{D_f}^{(j,r,s)}(v) = 1 \end{cases} \quad (96)$$

zaś analogicznie (90) funkcję wyjścia można zapisać następująco:

$$I_{O_s}^{(j,r)} = \lambda_k \left(I_{D_f}^{(j,r,s)}(v), A^{(j,r,s)}(v) \right) \quad (97)$$

$$\text{gdzie } \lambda_k = \begin{cases} 0, & \text{gdy } A^{(j,r,s)}(v) = A_0^{(j,r,s)} \\ I_{O_s}^{(j,r)}, & \text{gdy } A^{(j,r,s)}(v) = A_1^{(j,r,s)} \end{cases} \quad (98)$$

Schematycznie taki automat przedstawiono na Rys. 9.



Rys. 9. Automat – wersja rozszerzona

Wreszcie agregację można przeprowadzić dla całego badanego systemu zarządzania. Również tu możliwe jest zachowanie automatu w postaci elementarnej (jak dla operacji), lub częściowo zagregowanej (jak dla komórki), jednak bardziej interesujące wydaje się stworzenie jednego quasi-automatu dla całego systemu. Alfabet wejściowy takiego automatu będzie zbiorem identyfikatorów wszystkich operacji dostawczych dla danego systemu:

$$I_D = \left\{ I_{D1}^{(1,1,1,1)}, \dots, I_{Df}^{(i,j,r,s)}, \dots, I_{Dp}^{(k,l,m,n)} \right\} \quad (99)$$

Alfabetem wyjściowym będzie zbiór identyfikatorów wyników wszystkich operacji systemu.

$$I_O = \left\{ I_{O1}^{(1,1,1)}, \dots, I_{Os}^{(i,j,r)}, \dots, I_{On}^{(k,l,m)} \right\} \quad (100)$$

Zbiór stanów będzie się składał ze stanów wszystkich operacji całego systemu:

$$A = \left\{ A_0^{(1,1,1,1)}, A_1^{(1,1,1,1)}, \dots, A_0^{(i,j,r,s)}, \dots, A_1^{(k,l,m,n)} \right\} \quad (101)$$

Częstotliwość kroku określona jest przez najmniejszą okresowość operacji w całym systemie.

Podobnie jak poprzednio funkcja przejścia ma postać:

$$A^{(i,j,r,s)}(v+1) = \delta_s \left(I_{D_f}^{(i,j,r,s)}(v), A^{(i,j,r,s)}(v) \right) \quad (102)$$

gdzie:

$$\delta_s = \begin{cases} A^{(i,j,r,s)} = A_0^{(i,j,r,s)} = 0, & \text{gdy } \bigvee_{i=1}^k \bigvee_{j=1}^l \bigvee_{r=1}^m \bigvee_{s=1}^n \bigvee_{f=1}^p I_{D_f}^{(i,j,r,s)}(v) = 0 \\ A^{(i,j,r,s)} = A_1^{(i,j,r,s)} = 1, & \text{gdy } \bigwedge_{i=1}^k \bigwedge_{j=1}^l \bigwedge_{r=1}^m \bigwedge_{s=1}^n \bigwedge_{f=1}^p I_{D_f}^{(i,j,r,s)}(v) = 1 \end{cases} \quad (103)$$

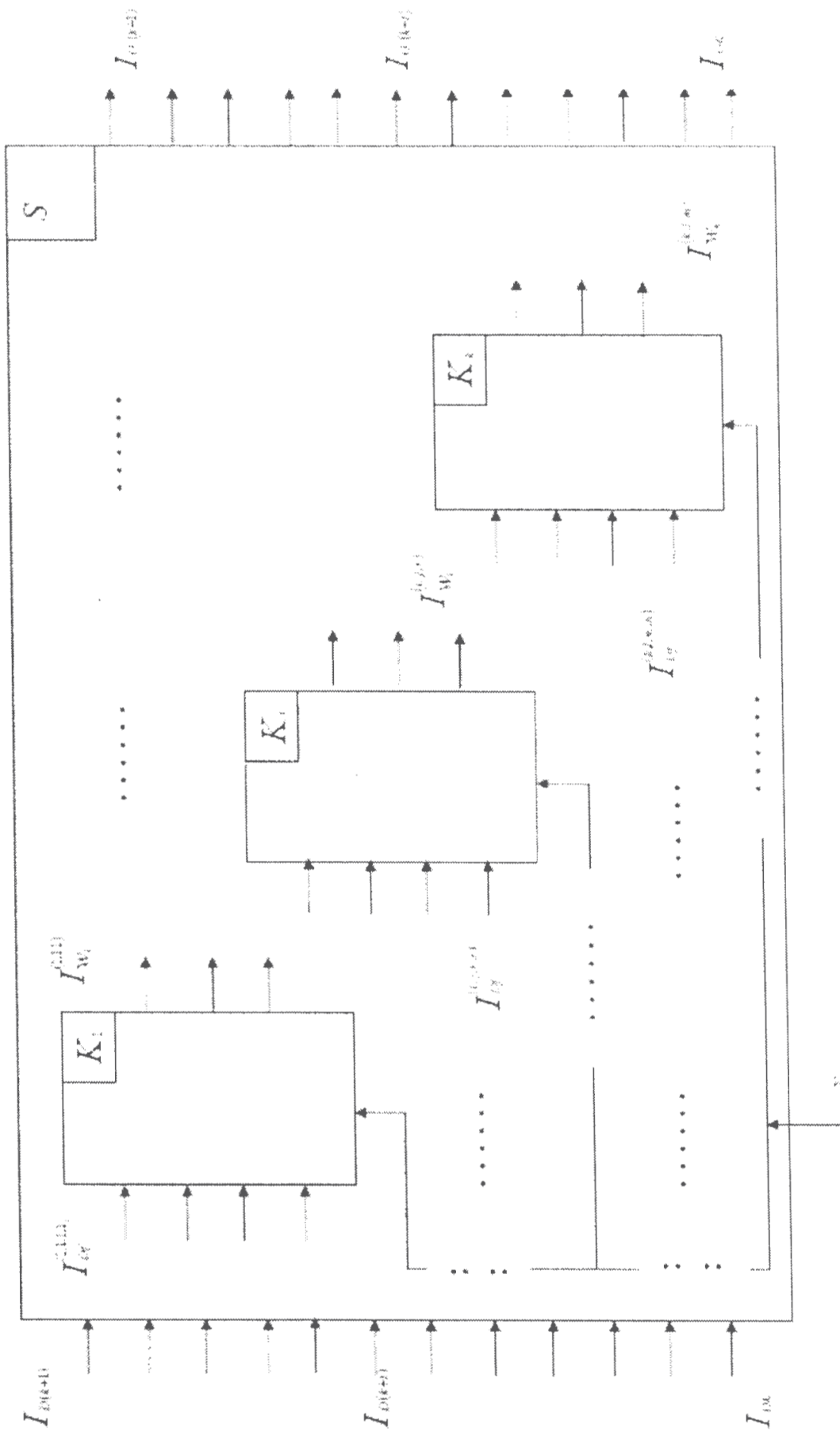
oraz funkcja wyjścia:

$$I_{ws}^{(i,j,r)}(v) = \lambda_s \left(I_{D_f}^{(i,j,r,s)}(v), A^{(i,j,r,s)}(v) \right); \quad (104)$$

gdzie:

$$\lambda_s = \begin{cases} 0, & \text{gdy } A^{(i,j,r,s)}(v) = A_0^{(i,j,r,s)} \\ 1, & \text{gdy } A^{(i,j,r,s)}(v) = A_1^{(i,j,r,s)} \end{cases} \quad (105)$$

Schematycznie ten automat przedstawiono na Rys. 10. Na zerowym kroku automat jest w stanie początkowym. Krok zerowy jest zsynchronizowany z czasem rzeczywistym i odpowiada momentowi początkowemu modelowanego okresu (np. dla roku będzie to godzina 0:00 1 stycznia). Wielkość kroku odpowiada najmniejszej okresowości operacji (aktualne oprogramowanie dopuszcza minimalny krok = 1 godzina). W momencie startowym do odpowiednich wejść zostają wprowadzone wielkości z alfabetu wejściowego od wszystkich dostawców zewnętrznych (ponieważ nie są wyspecyfikowani na osi czasu) oraz tych dostawców wewnętrznych, którzy w tym momencie mogą przekazać wyniki wykonania operacji (dla kroku zerowego w celu ustalenia tych dostawców modelowanie przeprowadza się z odpowiednim cofnięciem się na osi czasu do roku poprzedniego).



Rys. 10. Automat – wersja pełna

W toku modelowania następują kolejne kroki automatu. W przypadku spełnienia warunku kompletności dostawców wynik operacji zostaje wstrzymany o 1 krok (lub więcej kroków, jeżeli jest zapamiętywany również czas niezbędny na wykonanie operacji), następnie przekazany odbiorcy (co odpowiada zmianie stanu automatu). W przeciwnym przypadku stan automatu pozostaje bez zmian.

Taka postać modelu umożliwi analizę struktury logicznej powiązań informacyjnych w badanym systemie dając w wyniku analogiczny rezultat co i przy modelu statycznym (nie istnieje więc w tym przypadku konieczność uprzedniego uzyskania poprawnej sieci). Ponadto model daje pewne możliwości analizy dynamiki systemu, co pozwala wykryć niektóre objawy niedomagań (jak np. brak synchronizacji w czasie, dysfunkcjonalność, wąskie gardła i szereg innych – patrz Rozdział IV).

Spróbujmy teraz porównać model statyczny i dynamiczny. Najpierw zrobimy to na drodze intuicyjnej. Wyobraźmy sobie, że badanym obiektem jest ruchliwa dzielnica miejska i wykonaliśmy w nocy jej kolorowe zdjęcie nastawiając czas naświetlania np. 6 godzin (badany okres). Zakładamy, że każdy z poruszających się w tym czasie pojazdów miał inny kolor świateł. Na zdjęciu uzyskamy wszystkie drogi przebyte w tym czasie przez wszystkie pojazdy w obrębie badanego obszaru. Jest to właśnie ten wynik, który uzyskujemy w modelu statycznym. Jeżeli teraz zamiast aparatu fotograficznego weźmiemy kamerę filmową to poszczególne klatki taśmy utrwalały taki obraz, jaki uzyskujemy w kolejnym kroku automatu w modelu dynamicznym (kolejny numer klatki będzie kolejnym krokiem automatu). Podobnie jak w filmie tak i w tym modelu uzyskujemy efekt pseudodynamiki.

Przeprowadźmy teraz porównanie na drodze analitycznej. Niech skończona ilość (zbiór T) kroków v automatu została ponumerowana od 1 do z . Wówczas zależność uaktywnienia poszczególnych elementów modelu statycznego w czasie można ująć kwantyfikatorem egzystencjalnym

$\bigvee_{v=1}^z T(v)$ wprowadzonym do (84):

$$\bigvee_{v=1}^z T(v) \underset{0}{\bigwedge} I_O \underset{0}{\bigvee} I_{O_D} \underset{0}{\bigvee} I_{O_w} \Rightarrow \left(I_{O_D} = I_O \right) \wedge \left(I_{O_w} = I_O \right) \quad (106)$$

Nietrudno sobie wyobrazić, że jeżeli wszystkie powiązania zostaną wykryte przed upływem modelowanego okresu, to w tym momencie powyższy warunek zostanie zadość uczyniony. W przeciwnym przypadku modelowanie przeprowadza się do końca ($v = z$).

Dla modelu dynamicznego użyjemy kwantyfikatora ogólnego

$\bigwedge_{v=1}^z T(v)$, sprowadzając (102), (104) do postaci:

$$\begin{aligned} \bigwedge_{v=1}^z T(v) \quad \bigvee_{i,j,r,s=1}^{k,l,m,n} A(i,j,r,s) &\Rightarrow \left[\left(\bigvee_{i=1}^k \bigvee_{j=1}^l \bigvee_{r=1}^m \bigvee_{s=1}^n \bigvee_{f=1}^p I_D(i,j,r,s,f)=0 \right) \Rightarrow \right. \\ &\Rightarrow \left. \left(\bigvee_{i,j,r,s=1}^{k,l,m,n} A(i,j,r,s) \right) \right] \wedge \left[\sim \bigvee_{i,j,r,s=1}^{k,l,m,n} I_O(i,j,r,s) \right] \vee \end{aligned} \quad (107)$$

$$\vee \left[\left(\bigwedge_{i=1}^k \bigwedge_{j=1}^l \bigwedge_{r=1}^m \bigwedge_{s=1}^n \bigwedge_{f=1}^p I_D(i,j,r,s,f)=1 \right) \Rightarrow \right.$$

$$\left. \Rightarrow \left(\bigvee_{i,j,r,s=1}^{k,l,m,n} A(i,j,r,s)=1 \right) \right] \wedge \left[\bigvee_{i,j,r,s=1}^{k,l,m,n} I_O(i,j,r,s) \right]$$

Biorąc pod uwagę, że $[A(i,j,r,s) = 1]$ odpowiada $I_{O_W}(i,j,r,s)$,

czyli $\bigwedge_O I_{O_W}$, zaś $[A(i,j,r,s) = 0] \equiv \sim [A(i,j,r,s) = 1]$ oraz

w zależności od kroku $I_D(i,j,r,s) = I_{O_D} \vee I_{O_W}$, można udowodnić równoważność powyższych opisów modelu statycznego i dynamicznego.

Notabene, przy modelowaniu na komputerze postępuje się właśnie zgodnie z takim algorytmem. Jedyńa różnicą może (ale nie musi) polegać na zadeklarowaniu powiązań danych wejściowych względem $T(v)$. W modelu statycznym zależność ta może być zupełnie przypadkowa (jeśli dane

wejściowe są przetwarzane wg kolejności ich wprowadzania), lub też ukierunkowana na minimalizację czasu modelowania, co uzyskuje się po odpowiednim wstępnym przetworzeniu danych wejściowych. Ale nic nie stoi na przeszkodzie, by symulacja modelu statycznego odbywała się wg tej samej zależności $T(v)$, co i dla modelu dynamicznego, dla którego jest ona z góry zadana.

Z powyższego wynikają dwa podstawowe wnioski:

- w opracowanym modelu dynamicznym, który daje się sprowadzić do modelu statycznego, nie udało się wyeliminować niektórych wad typowych dla tego ostatniego. Między innymi nie istnieje możliwość uchwycenia zmian zachodzących w tym samym kroku (odpowiednik procesów przejściowych), a więc nie jest on faktycznie modelem dynamicznym.
- modelowanie, wg $T(v)$ narzuconego przez badany obiekt nie daje możliwości skrócenia czasu modelowania.

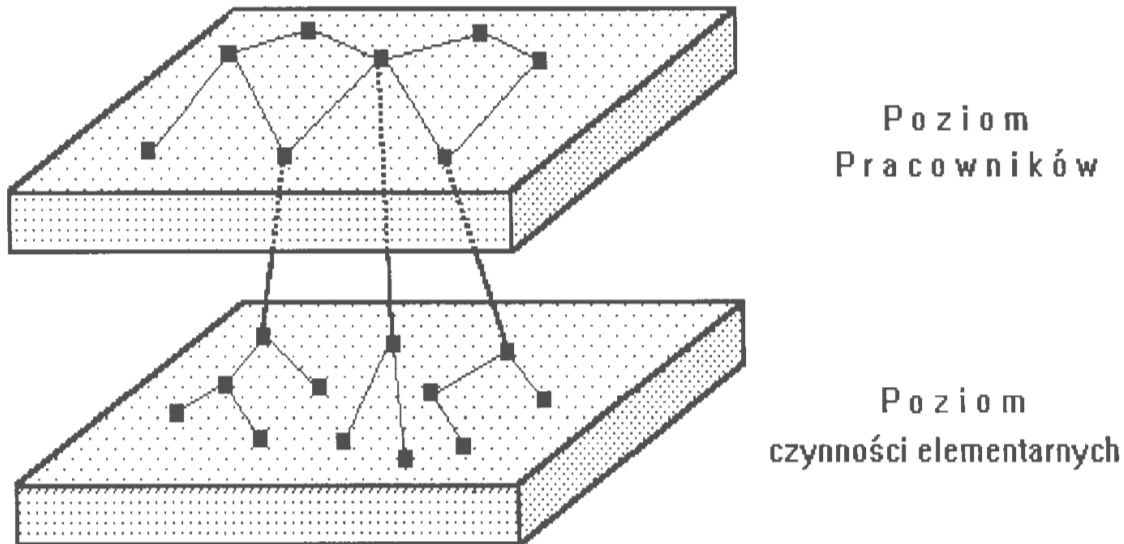
Reasumując: ta wersja modelu posiadała wiele zalet, jednak również i dwa istotne mankamenty: model był statyczny oraz nie uwzględniał tak ważnego czynnika w zarządzaniu organizacją jak czynnik ludzki, wbrew słusznej zasadzie, że organizacja to przede wszystkim ludzie. Kto o tym zapomina może się znaleźć pewnego dnia w bardzo nieciekawej sytuacji.

Z tych przyczyn przeprowadzono pewne badania zmierzające do opracowania nowego modelu dynamicznego. Prowadzono je w dwóch podstawowych kierunkach - modyfikacji samego modelu i modyfikacji narzędzia opisu.

Na tej podstawie zostały opracowane również programy, które weszły w skład następnej wersji pakietu DIANA.

III.2. Dwupoziomowy model dynamiczny systemu informacyjnego zarządzania

W pierwszej kolejności model został rozbudowany o drugi poziom - sieć powiązań pomiędzy pracownikami (rys. 11.):



Rys. 11. Dwupoziomowy model systemu informacyjnego zarządzania

Powiązania między pracownikami, np. Panem X i Panem Y, wynikają z powiązań zadań przez nich realizowanych. Jeżeli Pan X przekazuje wyniki swych zadań Panu Y, to powstaje problem: czy on je wykorzystuje, a jeżeli tak - to w którym ze swoich zadań (mamy wówczas powiązanie), czy jednak nie są one mu potrzebne (brak powiązania). Rozstrzygnięcie tego problemu i ustalenie jaki stan jest naprawdę stanowi zadanie pakietu DIANA na etapie diagnozy – patrz Rozdział IV. Widzimy więc, że mamy tu nie tylko luki poziome (powiązania między pracownikami), ale również pionowe, które wskazują jakie zadania realizuje dany pracownik.

Ten poziom posiada naturalną hierarchię: naczelny, jego zastępcy aż po szeregowego pracownika. Ta część metodyki (wraz algorytmami diagnostycznymi) została opracowana w ścisłej współpracy z Wydziałem Psychologii Uniwersytetu Warszawskiego i Zakładem Socjologii PAN.

Dwupoziomowy model był wykorzystywany w kilku następnych wersjach pakietu DIANA, jednak istotnym jego ograniczeniem było to, że odzwierciedlał on statyczny obraz funkcjonowania badanego systemu informacyjnego zarządzania. Można to obrazowo przedstawić w następujący sposób: fotografujemy nocą ze wzgórza duże miasto, nastawiając migawkę na kilka godzin. Na zdjęciu otrzymamy nie tylko nieruchome obiekty, ale również w postaci linii ślady poruszających się pomiędzy nimi pojazdów.

Brak będzie informacji, który z nich przejechał wcześniej, który jechał szybciej itd. Podobną sytuację mamy w naszym modelu - odzwierciedla on wszystkie powiązania w badanym systemie zarządzania w ciągu całego symulowanego okresu (jeden rok) tak jakby nastąpiły one w jednym momencie. Z tego powodu nie było np. możliwości identyfikacji tak częstego objawu w systemach zarządzania jak brak synchronizacji realizacji zadań. Chodzi w szczególności o przypadki, gdy zarówno dostawca informacji, jak też jego odbiorca działają zgodnie z wadliwymi przepisami (np. dostawca ma kończyć swoje zadanie 15-tego każdego miesiąca, zaś odbiorca - 10-tego każdego miesiąca). Są to nagminne przypadki w powiązaniach pomiędzy pionami, ponieważ z reguły obieg informacji jest w każdym projektowany niezależnie. W modelu statycznym nie było również możliwości uchwycenia zmian obciążenia stanowisk w czasie, ani też umiejscowienia wąskich gardeł na osi czasu.

W wyniku dalszych badań udało się stworzyć wariant istotnie dynamiczny modelu. Początkowo krokiem symulacji dynamicznej była jedna minuta. Ograniczyło to (ze względu na możliwości obliczeniowe ówczesnego sprzętu) wielkość symulowanej sieci. Z kolei przyjęcie za krok symulacji jeden dzień powodowało małą przydatność uzyskiwanych wyników. Ostatecznie zdecydowano się na nietrywialne rozwiązanie - zmieniający się, w zależności od aktywności stanowiska, krok symulacji w przedziale od jednej godziny do jednego dnia.

Pierwszą narzucającą się myślą w kierunku udoskonalenia modelu jest pomysł zwiększenia częstotliwości kroku automatu. Istotnie, można sobie wyobrazić modelowanie dla tak małego okresu w ciągu którego w realnym obiekcie nie mogą zajść bardziej istotne zmiany w sieci powiązań. Jednak pociąga to za sobą konieczność uzyskania bardziej szczegółowych danych. Więcej nawet - będą one musiały być bardziej szczegółowe, niż zwyczajowy i ustalony podział czynności (np. operację wykonywaną co 1 godzinę należałoby sztucznie rozbić na drobniejsze części według bliżej nieokreślonego subiektywnego kryterium). Pomijając sprawę (niebagatelną) dodatkowych „szumów” i błędów takiego dokładniejszego „pomiaru” powstaje problem jak zrealizować taki model na komputerze. Zakładana wówczas górna granica wielkości obiektu na 100.000 węzłów mogłaby się okazać za małą, co przy znacznym zwiększeniu ilości kombinacji powiązań i zmniejszaniu czasu odpowiadającego krokowi (np. do 1 minuty) w ogóle uniemożliwiłoby modelowanie na jakimkolwiek komputerze. Notabene 1 minuta nie byłaby wystarczająco małą okresowością dla uchwycenia procesów przejściowych w takich obiektach jak np. sieć energetyczna.

Innym sposobem może być przyjęcie, jako podstawy modelu dynamicznego, grupy automatów o postaci elementarnej (czyli 1 - go wariantu, pominiętego w poprzednich rozważaniach). Pewnej modyfikacji wymaga jednak również sam automat elementarny. Rozpatrzmy to bardziej szczegółowo.

Automat ten, będący modelem dowolnej oddzielnie rozpatrywanej s -tej operacji, zapisany przy pomocy kwantyfikatorów, będzie miał następującą postać:

$$\bigwedge_{v=1}^z T(v) \bigvee_s A(s) \Rightarrow \left[\left(\bigvee_s \bigwedge_{f=1}^p I_D(s,f) = 0 \right) \Rightarrow \left(\bigvee_s A(s) = 0 \right) \right] \vee \quad (108)$$

$$\vee \left[\sim \bigvee_s I_{Q_s} \right] \vee \left[\bigvee_s \bigwedge_{f=1}^p (I_D(s,f) = 1) \Rightarrow \left(\bigvee_s A(s) = 1 \right) \right] \wedge \left[\bigvee_s I_{Q_s} \right]$$

Wprowadźmy teraz nowy warunek - w przypadku, gdy prawdziwy jest drugi człon implikacji automat zadziała (przekaze I_{Q_s}) nie czekając na następny krok i w ramach tego samego kroku uruchomi mechanizm drobniejszych kroków (zmieni skalę modelowania). Dla ustalenia uwagi założymy, że:

$$\bigwedge_{v=1}^z T(v) \equiv (cH(h)) \quad h = 1, \dots, \gamma \quad (109)$$

gdzie:

c - jest właśnie nową skalą modelowania $\left(c = \frac{1}{\gamma} \right)$

h - kolejny numer nowego kroku wewnątrz starego

Inaczej mówiąc zakładamy, że każdy z kroków v składa się z jednakowej ilości kroków h .

By zrealizować to zamierzenie będziemy traktowali drugi człon alternatywy (108) jako poprzednik nowej implikacji, której następnik ma postać:

$$L \Rightarrow \left[\bigvee_{h=1}^{\gamma} H(h) \wedge \left(\bigvee_s \mathbf{V}_1 I_{O_s} = \bigvee_{f=1}^p I_D(f) \right) \right] \quad (110)$$

Korzystając z praw rachunków kwantyfikatorów wyrażenie (108) + (110) można nieco uprościć sprowadzając do postaci:

$$\bigwedge_{v=1}^z T(v) \bigvee_s \mathbf{V}_1 A(s) \Rightarrow \left[\bigvee_s \mathbf{V}_1 A(s) = 1 \right] \wedge \left[\bigvee_s \mathbf{V}_1 I_{O_s} \right] \Rightarrow \quad (111)$$

$$\Rightarrow \left[\bigvee_{h=1}^{\gamma} H(h) \wedge \left(\bigvee_s \mathbf{V}_1 I_{O_s} = \bigvee_{f=1}^p I_D(f) \right) \right]$$

Automat kroczy dużymi krokami dopóki nie zostaje uaktualniony (zmiana stanu) i wówczas przechodzi na mniejsze kroki. Sprawa staje się bardziej skomplikowana, ale jednocześnie i bardziej ciekawa, gdy podobny tok rozumowania zastosujemy do całego badanego obiektu. W tym wypadku zamiast jednego quasi automatu konstruujemy system składający się z tylu automatów elementarnych ile cały obiekt posiada operacji do wykonania. Utrzymamy w sile założenie o jednakowej i jednoczesnej skali dla całego systemu (109). Wówczas odpowiednio zredukowana postać analityczna (108) + (110) będzie następująca:

$$\begin{aligned} \bigwedge_{v=1}^z T(v) \bigvee_{i,j,r,s=1}^{k,l,m,n} A(i,j,r,s) &\Rightarrow \left[\bigvee_{i,j,r,s=1}^{k,l,m,n} A(i,j,r,s) = 1 \right] \wedge \left[\bigvee_{i,j,r,s=1}^{k,l,m,n} I_O(i,j,r,s) \right] \Rightarrow \\ &\Rightarrow \left[\bigvee_{h=1}^{\gamma} H(h) \wedge \left(\bigvee_{i,j,r,s=1}^{k,l,m,n} I_O(i,j,r,s) = \bigvee_{i,j,r,s,f=1}^{k,l,m,n,p} I_D(i,j,r,s,f) \right) \right] \quad (112) \end{aligned}$$

Z powyższego wynika, że jeśli którykolwiek automat zmieni stan, to cały system zmienia skalę, przechodząc na mniejszy krok, umożliwiając przekazanie wielkości wyjściowych odpowiednim automatom. Jeżeli spowoduje to zmianę stanu następnych automatów, to w którymś z następnych małych kroków nastąpi dalsze przekazanie wielkości wyjściowych itd.

Jest oczywiste, że jeżeli modelujemy jakiś obiekt rzeczywisty w którym istnieją dodatkowe uwarunkowania w czasie uaktywnienia poszczególnych jego elementów, to musimy je wprowadzić również do modelu w postaci odpowiedniej implikacji. Na przykład, jeżeli S -ta operacja nie może być wykonana wcześniej niż czas t (któremu odpowiada jakieś konkretne v_s^*), to w środkowy człon wyrażenia (112) wprowadzamy dodatkowo

$$\mathbf{L} \left[\left(\bigvee_{i,j,r,s=1}^{k,l,m,n} v^*(i,j,r,s) \geq v \right) \Rightarrow \mathbf{K} \right] \quad (113)$$

odpowiednio rozszerzając zasięg kwantyfikatora ogólnego dla $T(v)$. Analogicznie możemy postąpić i dla mniejszych kroków $H(h)$. Wówczas przedstawiony w (112) model w ciągu jakiegoś dużego kroku v (np. odpowiadającego dobowi) wykona wszystkie operacje odnoszące się do tego kroku w takcie małych kroków h (np. odpowiadających godzinom). Jeżeli takich operacji w tym dużym kroku nie ma, lub gdy w ogóle żaden z automatów nie zostanie uaktywniony (nie spełnia pierwszego warunku o kompletności dostawców), to system natychmiast przejdzie do następnego dużego kroku. Nie trudno sobie wyobrazić podobny tok postępowania, gdy i mały krok h okaże się „za duży” - możemy wówczas i ten krok podzielić na mniejsze itd., aż do zadośćuczynienia wymaganiom modelowanego obiektu. Zauważmy, że takie podejście jest znacznie bardziej korzystne, niż gdybyśmy od razu przyjęli za v najmniejszy krok. Przypuśćmy, że dla konkretnego obiektu taki krok odpowiada jednej minucie (przy badaniach obiektów rzeczywistych mieliśmy kilkakrotnie tego rodzaju sytuacje). Wówczas dla badanego okresu 1 rok, musielibyśmy zaplanować $T(v) = 525.600$ kroków i w zdecydowanej większości tych kroków nic by się nie działo. Przyjmując zaś zmienną skalę modelowania planujemy $T(v) = 365$

(V odpowiada jednej dobie) i tylko od czasu do czasu system w miarę potrzeby zmieniałby tę skalę.

Przy okazji warto zauważyć, że w podobny sposób jak były wprowadzone uwarunkowania kompletności dostawców oraz czasu uaktywnienia, można uwzględnić wszelkie inne uwarunkowania i ograniczenia zależnie od konkretnego obiektu (w tym również czas niezbędny na wykonanie operacji, o którym była już mowa).

Wystarczy teraz usunąć ograniczające założenie o jednoczesnej zmianie skali dla całego systemu dając taką możliwość indywidualnej zmiany skali każdemu z automatów elementarnych, by w wyniku uzyskać model istotnie dynamiczny. Można to uczynić wprowadzając do (112) odpowiednią zmianę w następniku ostatniej implikacji, a mianowicie:

$$\begin{aligned} L \Rightarrow & \left[\bigvee_{i,j,r,s,h=1}^{k,l,m,n,\gamma} \left(H(i,j,r,s,h) \vee \bigvee_{g=0}^n C(g) H(i,j,r,s,h) \right) \wedge \left(\bigvee_{i,j,r,s=1}^{k,l,m,n} I_O(i,j,r,s) = \right. \right. \\ & \left. \left. = \bigvee_{i,j,r,s,f=1}^{k,l,m,n,p} I_D(i,j,r,s,f) \right) \right] \end{aligned} \quad (114)$$

gdzie: $C(i,j,r,s,g)$ umożliwia ewentualne dalsze indywidualne „rozdrabnianie”skali (od c_0 do c_N).

Tu warto jeszcze raz na chwilę powrócić do sprawy zbiorów wielokrotnie indeksowanych, by uświadomić sobie w pełni złożoność postawionego zadania - symulacji zaproponowanego wyżej modelu.

Będzie to bardziej oczywiste, jeżeli analogicznie do (80) rozpatrzmy zbiór kroków H :

$$H(i,j,r,s,h) \Leftrightarrow |K_i, Z_j, P_r, O_s, h| \quad (115)$$

Powyższa zależność wskazuje na skład identyfikatora h - tego kroku s - tej operacji r - tego podzadania j - tego zadania i - tej komórki (identyfikator systemu S , jednakowy dla wszystkich zależności, został pominięty). Mówiąc obrazowo każda operacja każdego podzadania itd. ma swoje własne h - zmieniające się od 1 do jakiejś własnej γ i zmiany te mogą następować niezależnie w różnych operacjach (celowo operujemy h

a nie v , ponieważ $c_0 h = T$, $c_1 h = v$, $c_2 h = h$, itd.; dla v pozostawiamy jego poprzednie znaczenie - jak dla klasycznego automatu). Np. $h = 05$ (piąty krok). Dla operacji 03 podzadania 07 zadania 02 komórki 08 identyfikator kroku 08.02.07.03.05, zaś operacji 04 tego samego podzadania, zadania i komórki 08.02.07.04.05, i będą to zupełnie różne i niezależne od siebie kroki (np. w różnych skalach c_g).

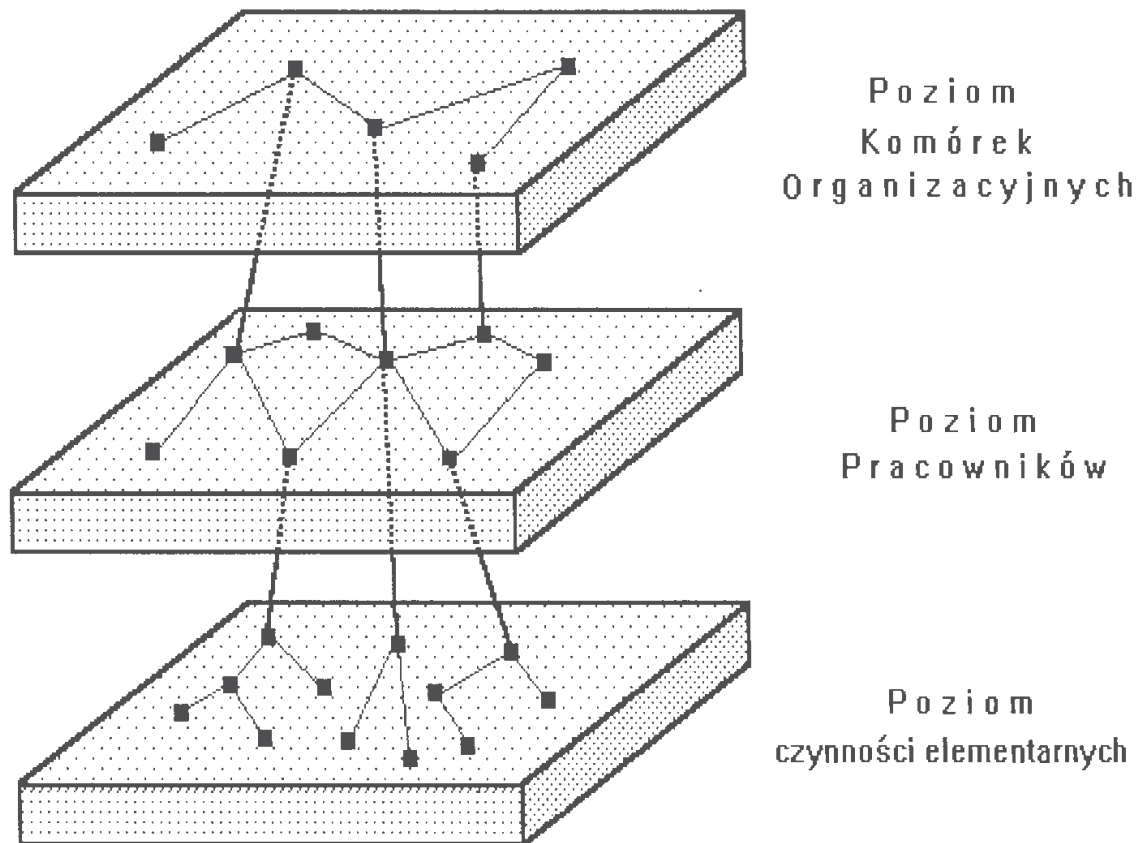
Powstaje pytanie - jak zrealizować taki model? Pomijając problem gigantycznej wprost pamięci niezbędnej w przypadku dużego obiektu oraz samej organizacji przetwarzania danych, pozostanie nierozwiązanym zagadnieniem modelowania sytuacji jednoczesnego uaktywnienia wielu automatów elementarnych (przypadek chyba dość częsty w obiekcie rzeczywistym). Zaakceptowanie rozwiązania obecnie powszechnie stosowanego w technice komputerowej, a mianowicie rozdzielenie w czasie, przekreśliłoby samą ideę modelu istotnie dynamicznego sprowadzając go do pierwotnej postaci modelu pseudo-dynamicznego.

Dla zupełnie małych obiektów rzeczywistych można byłoby spróbować zrealizować model na maszynie hybrydowej (jednoczesne przebiegi „procesów przejściowych” byłyby realizowane na części analogowej, zaś kontrola poprawności logicznej pełnej sieci i jej analiza - na części cyfrowej). Jednak wydaje się, że dla tak małych obiektów można wprowadzać usprawnienia bez użycia jakichkolwiek technik komputerowych i nie te obiekty są przedmiotem naszych zainteresowań.

Pozostaje więc oczekiwanie na zjawienie się nowej generacji komputerów („megaprocessorowych”), które umożliwiłyby realizację zaproponowanego modelu istotnie dynamicznego. By jednak nie czekać bezczynnie dokonamy następnej modyfikacji modelu. Stało się to konieczne również z tego powodu, że badania w kolejnych dużych obiektach rzeczywistych wykazały następną słabą stronę dwupoziomowego modelu, a mianowicie zaistniały poważne trudności przy próbach wdrażania usprawnień opracowanych na modelu. Brak było jednoznacznego przejścia od struktury istniejącej do struktury zmodyfikowanej. Szczególnie dawało to się we znaki przy bardzo dużych obiektach, gdzie ilość zmian była wielka. Mankamentem był również brak możliwości sprawdzania na modelu różnych propozycji zmian struktury organizacyjnej systemu informacyjnego zarządzania. Wówczas to doszliśmy do bardzo ważnego wniosku: wadliwa struktura organizacyjna jednoznacznie powoduje wadliwe funkcjonowanie systemu informacyjnego zarządzania. Wynikiem próby rozwiązania tych problemów było opracowanie następnej wersji modelu.

III.3. Trzy poziomowy polihierarchiczny model systemu informacyjnego zarządzania

W nowej wersji rozszerzono model o trzeci poziom (Rys. 12.), odzwierciedlający strukturę organizacyjną [91]. Warto chwilę uwagi poświęcić hierarchii tej struktury. Otóż, hierarchia ta odpowiada hierarchii struktury organizacyjnej rzeczywistego obiektu, jednak w modelu



Rys. 12. Trzy poziomowy model systemu informacyjnego zarządzania

jest ona przedstawiana w postaci inkluzji: na szczycie znajduje się cały obiekt, który zawiera w sobie poszczególne pion, te z kolei zawierają np. wydziały itd. Można to obrazowo przedstawić w następujący sposób (patrz Rys. 31.): cały badany obiekt jest jakby wielką skrzynią wewnątrz której znajdują się mniejsze skrzynki - pion, wewnątrz których są jeszcze mniejsze skrzynki - wydziały itd., zaś na samym dole hierarchii znajdują się najmniejsze skrzyneczki - stanowiska jednoosobowe. To założenie jest niezbędne i w praktyce łatwo je spełnić - dowolne stanowisko wieloosobowe dzielimy na tyle stanowisk jednoosobowych, ile osób jest w nim zatrudnionych. Dzięki temu możemy jednoznacznie związać w modelu

poziom komórek organizacyjnych z poziomem pracowników: po wprowadzeniu do modelu całej struktury organizacyjnej aż do poziomu stanowisk jednoosobowych wprowadzamy konkretne osoby w nich zatrudnione.

Obserwacja złożonych struktur organizacyjno - informacyjnych, a w szczególności tak doskonałego wzorca jakimi są bardziej złożone organizmy żywe oraz pewne doświadczenia uzyskane z badań na obiektach rzeczywistych, nasunęły ideę, która może stanowić podstawę przynajmniej czasowego rozwiązania problemu symulacji modelu istotnie dynamicznego. Jest ona poza tym sama w sobie wystarczająco ciekawa by poświęcić jej nieco uwagi.

Zwracając się ponownie najpierw do intuicji rozważmy następującą sytuację. Badanym obiektem jest złożona jednostka gospodarcza. Posiada ona zgodnie ze swoim schematem organizacyjnym określoną strukturę hierarchiczną od dyrektora naczelnego począwszy i na robotnikach niewykwalifikowanych kończąc (nazwijmy ją umownie strukturą podstawową). Jednocześnie na tę strukturę nakładają się inne struktury hierarchiczne - organizacji społecznych (np. związków zawodowych, gdzie prezesem może być robotnik niewykwalifikowany), politycznych, powiązań nieformalnych (dobry znajomy „mocnego”, znajomy tego znajomego itd.) i wiele, wiele innych. Struktury te w określony sposób współżyją ze sobą tworząc coś co nazwiemy strukturą polihierarchiczną. Notabene zagadnienie mechanizmu współżycia takich struktur jest pasjonującym tematem, lecz w kontekście danego rozdziału nie jest tak istotne.

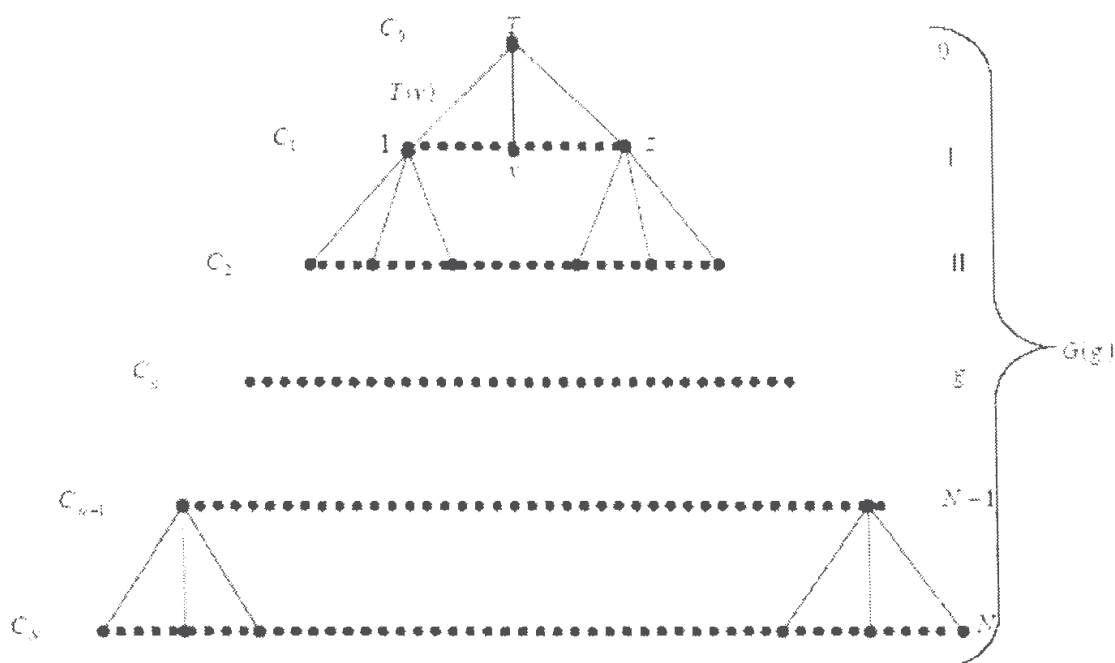
Struktury polihierarchiczne można zaobserwować w najróżniejszych dziedzinach. Posiadają ją również organizmy żywe z człowiekiem na czele - ma on oprócz systemu nerwowego (będącym strukturą podstawową funkcjonowania organizmu) szereg innych układów mających wyraźnie strukturę hierarchiczną - układ krwionośny, układ limfatyczny, układ kostny itd.

Łatwo zauważyć, że pojęcie struktury polihierarchicznej jest pojęciem fundamentalnym dla całego globalnego modelu systemu informacyjnego zarządzania (patrz Rozdział III.4). Na razie jednak wprowadźmy to pojęcie do rozpatrywanego modelu istotnie dynamicznego. Strukturami hierarchicznymi będą dla nas stopnie rozbicia „dużego” kroku na coraz drobniejsze w poszczególnych automatach. Za strukturę podstawową możemy przyjąć najdrobniejsze rozbicie możliwe w danym obiekcie.

Poszczególne automaty korzystają zawsze z pierwszego poziomu tego schematu i w miarę potrzeby z poziomów coraz niższych. Będzie to więc dość specyficzna struktura polihierarchiczna, w której poszczególne

struktury będą dokładną kopią całej, lub fragmentów struktury podstawowej. Byłaby to dość prymitywna struktura polihierarchiczna, gdyby nie to, że jest ona zmienna w czasie. Jednak przyjmując warunek stałej okresowości każdej z operacji można tę strukturę „zamrozić”. Warunek ten w badaniach praktycznych nie jest trudno zadośćuczynić - poprzez odpowiednie rozbitcie czynności na operacje (te same czynności, lecz o różnych okresowościach należy traktować jako różne operacje).

Teraz nasuwa się dość proste rozwiązanie - należy pogrupować operacje o takich samych strukturach hierarchicznych, a następnie realizować modelowanie wg podstawowej struktury hierarchicznej (Rys. 13). Miejsce, gdzie będzie ona miała dziury (żaden z automatów nie posiada struktury odpowiadającej tej części schematu) należy przy modelowaniu pomijać (przeskoczyć). Przypadki nakładania się kilku struktur wymagają oczywiście, jak na razie, modelowania z rozdzieleniem w czasie. Rozwiązanie takie nie jest zapewne idealnym, jednak można mieć nadzieję, że w obiektach rzeczywistych nie będą się zdarzały przypadki „złośliwe” - nakładania się zbyt wielkiej ilości struktur, uniemożliwiające realizację na komputerze. Największą zaletą polihierarchicznego modelu dynamicznego jest uwolnienie się od konieczności pełnego przeglądu operacji na każdym coraz malejącym kroku.



Rys. 13. Struktura polihierarchiczna.

Rozpatrzmy teraz pewne zależności analityczne proponowanego modelu. Struktura przedstawiona na Rys. 13 ma następującą postać:

$$\bigwedge_{v=1}^z T(v) \bigvee_{h=1}^{\gamma} H(h) \Rightarrow \bigwedge_{h=1}^{\gamma} H(h) \bigvee K \equiv \bigwedge_{v=1}^z T(v) \bigvee_{g=0}^N G(g) \Rightarrow \bigwedge_{g=0}^N G(g) \bigvee_{g=0}^N C(g) \quad (116)$$

Grupowanie automatów możemy zapisać następująco:

$$\bigwedge_{i,j,r,s=1}^{k,l,m,n} (A(i,j,r,s)=1) \bigvee_{g=0}^N C(g) \Rightarrow \bigvee_{g=0}^N C(g) \bigwedge_{i,j,r,s=1}^{k,l,m,n} A(i,j,r,s) \quad (117)$$

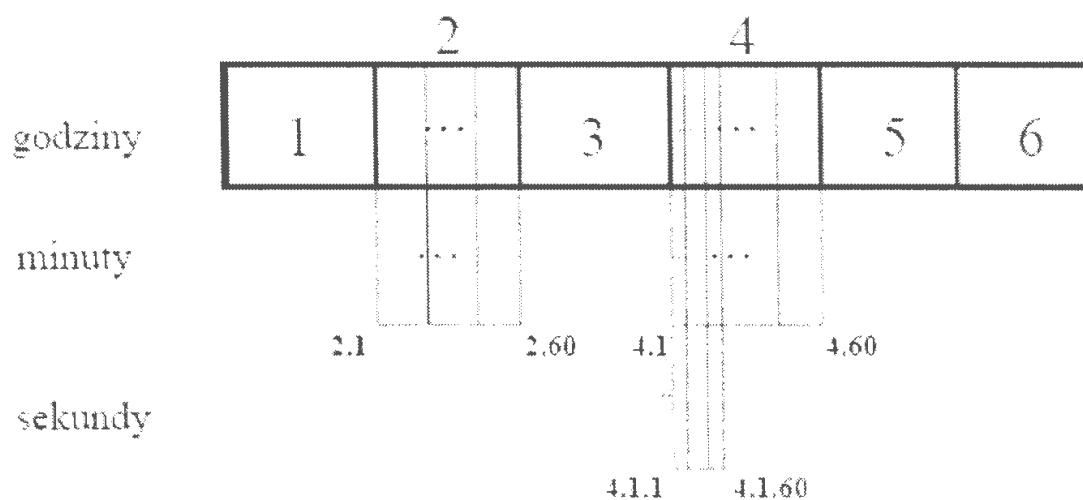
Z (116) i (117) wynika:

$$\bigwedge_{v=1}^z T(v) \bigvee_{g=0}^N G(g) \Rightarrow \bigwedge_{g=0}^N G(g) \bigvee_{i,j,r,s=1}^{k,l,m,n} A(i,j,r,s) \Rightarrow \bigwedge_{i,j,r,s}^{k,l,m,n} (A(i,j,r,s)=1) \bigvee_{g=0}^N G(g) \quad (118)$$

Możemy teraz w przedstawiony model istotnie dynamiczny (112) + (114) wprowadzić (118), by uzyskać polihierarchiczny model dynamiczny [72].

$$\begin{aligned} \bigwedge_{v=1}^z T(v) \bigvee_{i,j,r,s=1}^{k,l,m,n} A(i,j,r,s) &\Rightarrow \left[\bigvee_{i,j,r,s=1}^{k,l,m,n} A(i,j,r,s)=1 \right] \wedge \left[\bigvee_{i,j,r,s=1}^{k,l,m,n} I_0(i,j,r,s) \right] \Rightarrow \\ &\Rightarrow \left[\bigwedge_{i,j,r,s,h}^{k,l,m,n,\gamma} H(i,j,r,s,\gamma) \bigvee_{g=0}^N C(g) \right] \wedge \left[\bigvee_{i,j,r,s=1}^{k,l,m,n} I_0(i,j,r,s) = \bigvee_{i,j,r,s,f=1}^{k,l,m,n,p} I_D(i,j,r,s) \right] \end{aligned} \quad (119)$$

Działanie takiego modelu najlepiej zinterpretować ponownie na podanym wyżej przykładzie obserwacji ruchliwej dzielnicy miejskiej. Jeżeli w pierwszej, trzeciej, piątej i szóstej godzinie obserwacji nic się nie działo, zaś w drugiej godzinie nastąpił umiarkowany ruch, a na początku czwartej - natężony, to strukturę kroków modelu można przedstawić następująco (Rys. 14):



Rys. 14. Ilustracja zmiennego kroku symulacji

Czyli przy robieniu filmu częstotliwość klatek byłaby zależna od natężenia ruchu. Dla zaprojektowanego w ten sposób modelu potrzebna byłaby ogólna ilość kroków równa 183, wówczas gdy dla przypadku modelowania w najmniejszym kroku (sekundy) dla całego badanego okresu liczba kroków wyniosłaby 21.600.

Z powyższej ilustracji wynika oczywista analogia modelu dynamicznego do modelu polihierarchicznego. Różnice polegają jedynie na samym sposobie ich realizacji, przy czym ten drugi ma pod tym względem wyraźną przewagę.

Na zakończenie warto zwrócić uwagę na to, że jeżeli nie interesuje nas zależność w czasie, lecz tylko same powiązania między automatami (w naszym przykładzie nas interesowałaby rejestracja tylko w momentach jakiegoś ruchu pojazdów, a w pozostałym czasie nie robilibyśmy zdjęć w ogóle), to wystarczy z (119) usunąć poprzednik pierwszej implikacji. Uzyskamy wówczas polihierarchiczny model statyczny, analogiczny do przedstawionego wcześniej modelu statycznego (patrz 84) i (106), jednak różnice w tym przypadku są bardziej istotne – posiadając informację o powiązaniach „pionowych” możemy łatwo rozszerzyć ten model o poziom następny (Pracowników) i dalej (Komórek Organizacyjnych) aż do poziomu najwyższego (Celów) – patrz Rozdział III.4.

Z powyższego wynika również, że wszystkie algorytmy identyfikujące nieprawidłowości pełnej sieci, przedstawione w Rozdziale IV (m.in. brak synchronizacji w czasie, ślepa uliczka, odbiorcy dopisani, dysfunkcjonalność, wąskie gardła) są prawomocne także dla polihierarchicznego modelu.

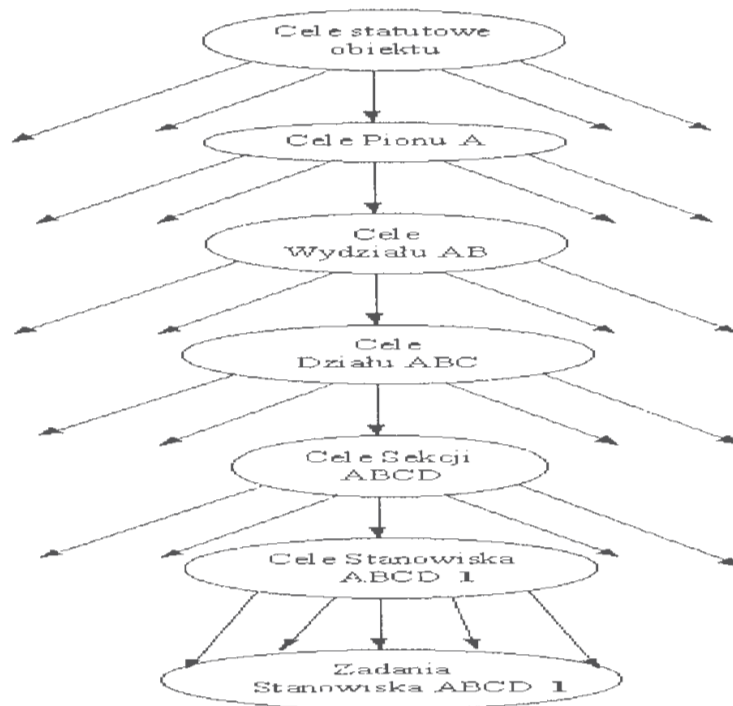
Ta wersja modelu otworzyła również możliwość nie tylko sprawdzania propozycji reorganizacji, ale też realizacji wspomaganego komputerowo projektowania struktur organizacyjnych od podstaw (patrz rozdz. V).

Wydawałoby się, że taki model spełnia już wszystkie warunki stawiane przy podejściu systemowym. Możemy drogą kolejnych przybliżeń uzyskać nader sprawny system zarządzania. Jednak nadal brak będzie odpowiedzi na istotne pytanie: po co? Czy istnieją racjonalne cele jego funkcjonowania i czy są one realizowane, oraz czy są na to odpowiednie środki? Inaczej mówiąc - do czego ma służyć nasz doskonale usprawniony obiekt i jakim kosztem? Te pytania szczególnie ostro wystąpiły po wejściu Polski na drogę gospodarki rynkowej. Dlatego zdecydowano się na następną modyfikację modelu.

III.4. Wielopoziomowy polihierarchiczny przestrzenny model systemu informacyjnego zarządzania

W następnej implementacji metody - pakiecie DIANA-9, powyższy model został rozszerzony o następny poziom, opisujący podstawowe cele działania badanego systemu informacyjnego zarządzania oraz odzwierciedlający zasoby, przeznaczone na realizację tych celów [10]. Hierarchia tego poziomu odpowiada dokładnie hierarchii struktury organizacyjnej, tworząc tzw. „drzewo celów” – niezwykle pomocne zarówno na etapie diagnozy, jak też projektowania systemu informacyjnego zarządzania. Bardziej szczegółowo będzie o tym mowa w rozdz. IV i V, tu zaś warto przyjrzeć się konstrukcji drzewa celów. Przedstawiono ją na Rys. 15. Na szczycie znajduje się misja (jeżeli została sformułowana) oraz cele statutowe całego badanego obiektu, a więc cele sformułowane w najważniejszych dokumentach powołujących daną organizację do życia (np. dla Narodowego Banku Polskiego te cele zostały umieszczone w odpowiedniej ustawie). Poniżej, zgodnie ze strukturą organizacyjną, mamy cele podrzędne – np. pionów, określone w regulaminie organizacyjnym. Konsekwentnie idąc w dół struktury i czerpiąc dane z obowiązujących w organizacji przepisów uzyskujemy cele wydziałów, działów lub sekcji. Wszystko to razem tworzy górną część drzewa celów. Cele podrzędne realizują w jakiejś części odpowiednie cele nadrzędne. Zakłada się, że w „zdrowym” drzewie celów cele podrzędne wyczerpują wszystkie związane z nimi cele nadrzędne. Cała wiedza przy budowie drzewa celów na tym etapie opiera się na obowiązujących w organizacji przepisach i dokumentacji. Tak się dzieje do

poziomu stanowisk (jednoosobowych), dla których kierunek tworzenia drzewa celów zmienia się na przeciwny - z dołu do góry.



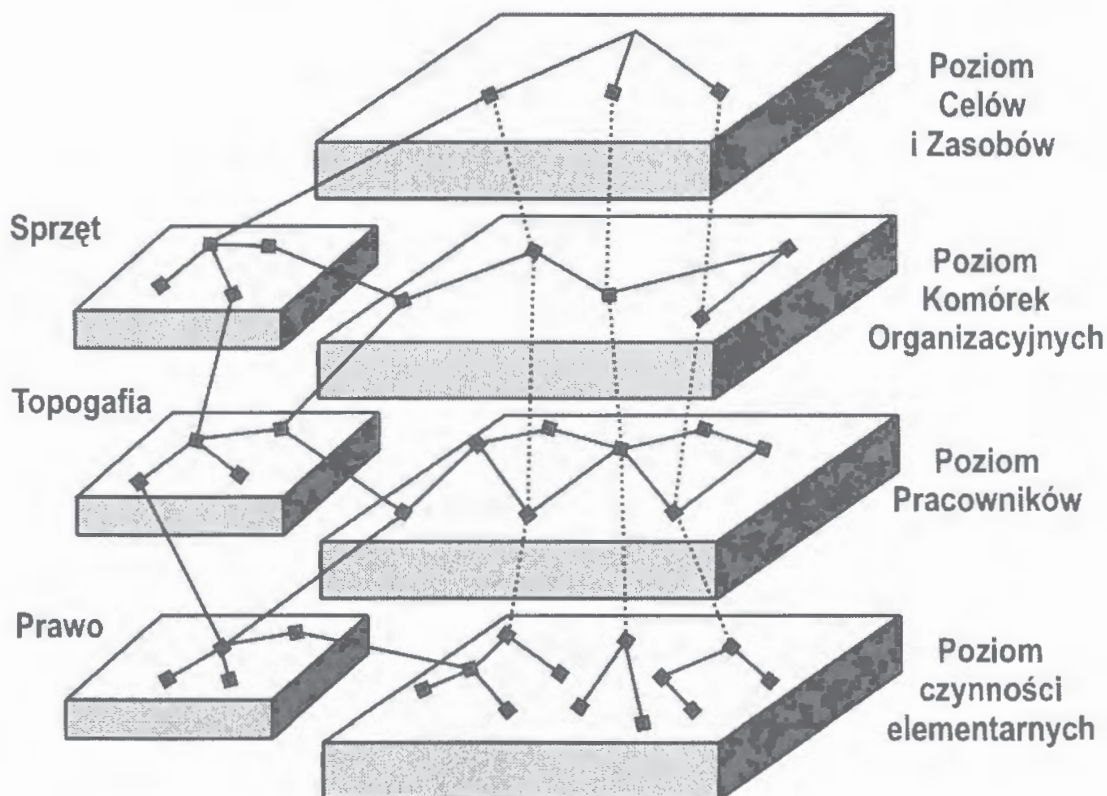
Rys. 15. Drzewo celów

Każde stanowisko jest opisane zestawem zadań rzeczywiście na nim wykonywanych (tworzą one „korzenie” drzewa celów), więc patrząc na te zadania (lub grupę zadań) musimy odpowiedzieć na pytanie: w jakim celu je wykonujemy? Te odpowiedzi utworzą listę celów stanowisk, które staramy się powiązać z odpowiednimi celami komórki nadrzędnej (np. sekcji). Powstaje więc niezwykle intrygujący problem: czy budowana na podstawie obowiązujących przepisów górna część drzewa celów będzie pasowała do jego podstawy, opartej na rzeczywiście realizowanych zadaniach? Tym zajmują się odpowiednie algorytmy diagnostyczne (patrz rozdział IV), które potrafią wykryć m.in. cel „papierowy” – sytuację, gdy nie jesteśmy w stanie wskazać ani osoby, ani zadania które realizowałoby ów cel – on istnieje tylko na papierze (czasami w bardzo ważnym dokumencie, np. statucie).

Być może w przyszłości będzie możliwa dalsza rozbudowa modelu [6]. W trakcie opracowywania najnowszej jedenastej wersji brana była pod uwagę możliwość uwzględnienia tzw. topografii – rozmieszczenie badanego obiektu w terenie, nasycenia komórek organizacyjnych w odpowiedni sprzęt, czy zmian w przepisach prawnych. Jednak już DIANA-9, jak się wydaje, spełnia podstawowe wymagania metody. Potwierdzeniem tego są

nie tylko wyniki badań na obiektach rzeczywistych, ale również „plon” zajęć ze studentami – w każdej grupie zadawałem „konkursowe” pytanie: „Czego jeszcze brakuje w tym modelu? Czym można go uzupełnić, by bardziej przybliżyć do modelu idealnego?” Szczególnie wartościowe były odpowiedzi studentów studiów zaocznych i wieczorowych, a więc osób mających doświadczenie zawodowe i praktyczne. Niestety, nie było ani jednej propozycji mającej ciekawą szansę na realizację! Jeżeli nawet była propozycja ciekawa lub oryginalna, to nie było sposobu ujęcia jej w postaci opisu formalnego a następnie oprogramowania.

Powstał więc niezwykle złożony przestrzenny model, który został jednolicie formalnie opisany za pomocą specjalnie opracowanego aparatu matematycznego: Q-algebry i dzięki temu oprogramowany, tworząc podstawę do osadzenia na nim pozostałych elementów pakietu. Ostatecznie najnowsza implementacja metody - pakiet DIANA-11, wykorzystuje model systemu informacyjnego zarządzania w postaci polihierarchicznej, wielopoziomowej, przestrzennej sieci powiązań informacyjnych [6]. Schematycznie przedstawiono ją na Rys. 16.:



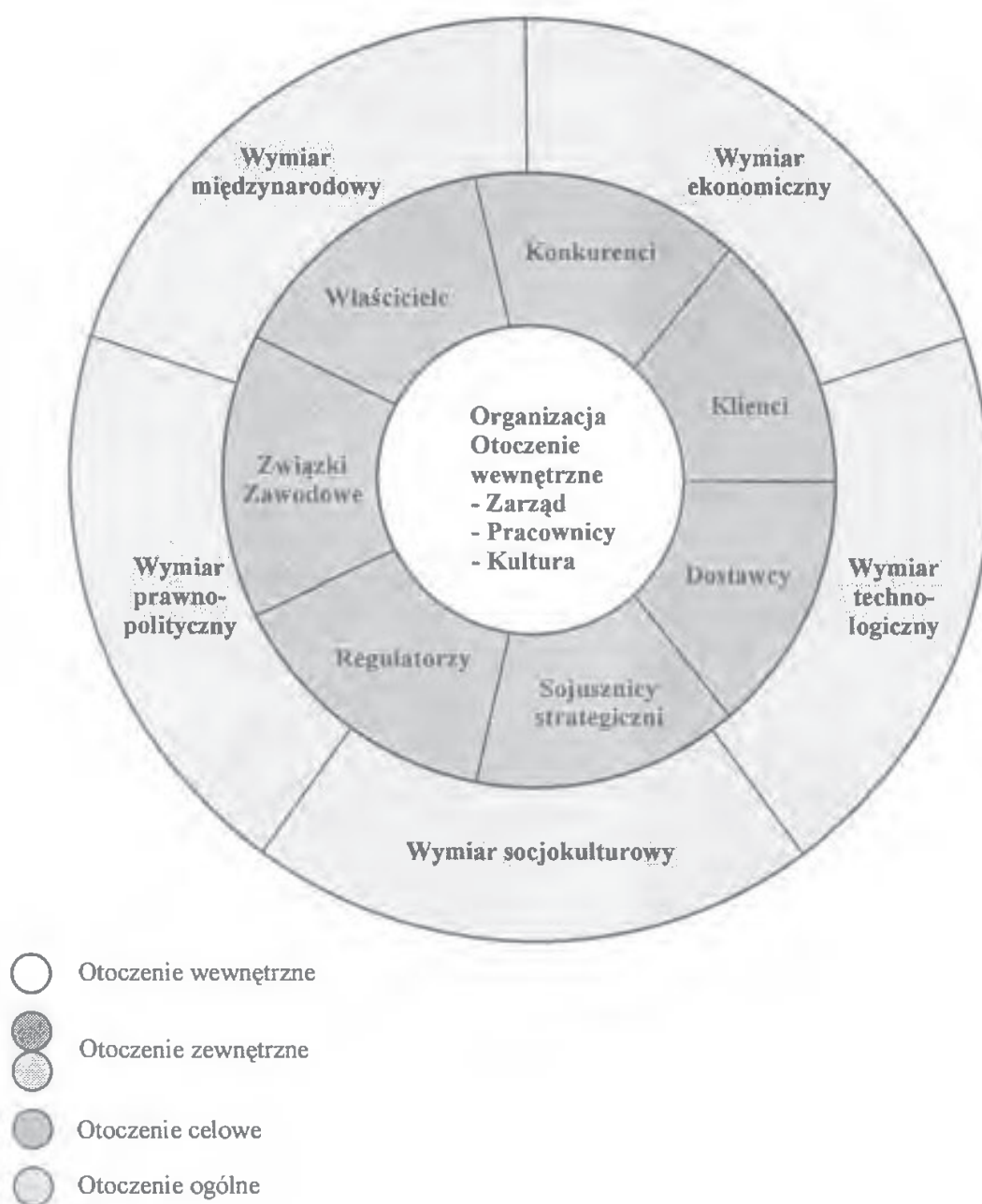
Rys. 16. Wielopoziomowy model systemu informacyjnego zarządzania

Na **najniższym poziomie** węzłami sieci są **czynności**, wykonywane przez poszczególnych pracowników badanego systemu informacyjnego zarządzania, zaś łukami wejściowe i wyjściowe informacje. Poziom ten posiada pewną hierarchię: zadania - podzadania - operacje (czynności elementarne). Zarówno węzły jak i łuki opisywane są szeregiem parametrów, wykorzystywanych nie tylko w diagnozie ale też przy projektowaniu struktury organizacyjnej.

Następny poziom zawiera zbiór **pracowników** (węzły sieci) - ich wzajemne powiązania (łuki sieci) wynikają z powiązań pomiędzy wykonywanymi przez nich czynnościami. Mamy więc nie tylko łuki poziome (dostawca – odbiorca informacji), ale również łuki pionowe, które wskazują jakie czynności realizuje dany pracownik. Poziom ten posiada własną hierarchię: np. prezes - jego zastępcy - kierownicy komórek - szeregowi pracownicy. Zarówno węzły jak i łuki opisywane są parametrami, wykorzystywanymi nie tylko w diagnozie, ale też przy projektowaniu struktury organizacyjnej.

Jeszcze **wyższym poziomem** jest sieć w której węzłami są **komórki organizacyjne**, zaś powiązania między nimi (łuki) wynikają z powiązań zatrudnionych w tych komórkach pracowników. Patrząc w dół – wynikają z powiązań realizowanych przez nich czynności. Tu również mamy łuki pionowe, wskazujące w jakiej komórce jest zatrudniony dany pracownik. Ten poziom posiada własną hierarchię: np. obiekt - pionowy - departamenty - wydziały - stanowiska. Zarówno węzły jak i łuki opisywane są parametrami, wykorzystywanymi nie tylko w diagnozie, ale też przy projektowaniu struktury organizacyjnej.

Wreszcie na **najwyższym poziomie** znajduje się sieć **celów i zasobów** badanego systemu informacyjnego zarządzania. Poziom ten posiada własną hierarchię: np. cele obiektu (statutowe) - cele pionów - cele departamentów - cele stanowisk. Zarówno węzły jak i łuki opisywane są parametrami (z których najważniejszymi są zasoby, czyli środki przeznaczone na realizację celów). Te parametry są wykorzystywane zarówno w diagnozie jak też przy projektowaniu struktury organizacyjnej. Powyższy model uzupełniają powiązania z otoczeniem. Jest to niezwykle istotna sprawa, dlatego starano się uwzględnić wszystkie najważniejsze sygnalizowane w literaturze aspekty relacji: obiekt – otoczenie. Ciekawą i wyczerpującą propozycję przedstawiono w [112], którą ilustruje Rys. 17.



Rys. 17. Organizacja i jej otoczenie
(źródło: [112])

Zgodnie z tą koncepcją w modelu wyodrębniono otoczenie wewnętrzne, w którym mogą znaleźć się nie tylko Zarząd, czy pracownicy (wszyscy ogółem – załoga), ale również inne komórki organizacyjne, które z różnych powodów nie są szczegółowo badane (np. stołówka pracownicza). Tutaj też znajdują się wszystkie istotne dla organizacji systemy informatyczne. W otoczeniu zewnętrznym, zarówno celowym jak też ogólnym, przewidziano nieograniczoną liczbę pozycji (w niektórych badanych przez nas obiektach

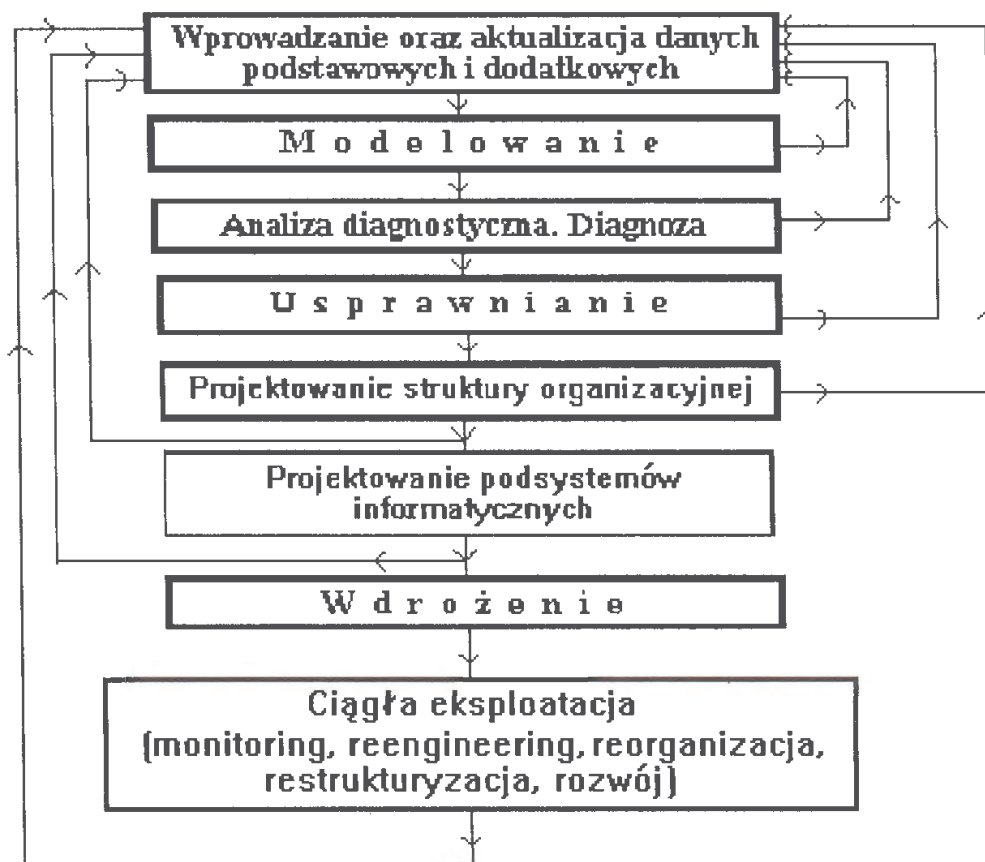
rzeczywistych sięgała ona rzędu kilkuset pozycji). Każdy element w otoczeniu wewnętrznym i zewnętrznym opisywany jest tak samo jak komórka organizacyjna przy założeniu, że realizuje jedno zadanie (tzw. „kontaktowe”, obejmujące całą jego działalność). Dzięki temu możliwe jest uwzględnienie nie tylko samych kontaktów informacyjnych z otoczeniem, ale również ocena ich jakości (poprzez parametry łuków), przydatna na etapie diagnozy i projektowania (patrz Rozdz. IV, V).

Trudno nie docenić wagi problemu dokładnego uwzględnienia relacji z otoczeniem. Dobrze zaprojektowany system informacyjny zarządzania powinien nie tylko sprawnie obsługiwać zarząd, ale powinien też z wyprzedzeniem dostarczać mu te informacje z otoczenia, które mogą mieć istotny wpływ na losy organizacji [81].

Na początku badań na nowym obiekcie model jest pusty. Można go wyobrazić jako wielki nowoczesny magazyn, w którym poszczególne hale zostały podzielone na boksy, wewnątrz których ustawiono wielopoziomowe stelaże – wszystko precyzyjnie oznakowane i powiązane siecią komunikacyjną, tak by każdy najdrobniejszy nawet element przyjmowany do magazynu natychmiast znalazł się tam gdzie powinien.

Blok wprowadzania danych (patrz Rys. 18.) pakietu DIANA-11 umożliwia wypełnienie powyższego modelu danymi z konkretnego obiektu. Proces ten zaczyna się od wprowadzenia aktualnie istniejącej struktury organizacyjnej. Równolegle wprowadzane są cele poszczególnych komórek organizacyjnych aż do stanowisk jednoosobowych włącznie. W tym momencie możemy jednoznacznie przypisać konkretną osobę do tego stanowiska. Mając już konkretnego Pana Kowalskiego możemy zapytać go, co on robi na tym stanowisku, jakie wykonuje zadania. Po wprowadzeniu tych danych pozostaje tylko dowiedzieć się skąd on czerpie informacje, by móc wykonać swoje zadania (dostawcy) oraz komu przekazuje wyniki realizacji tych zadań (odbiorcy). W ten sposób model systemu informacyjnego zarządzania, będący na początku jedynie szkieletem, teraz staje się modelem konkretnego badanego obiektu. I tak jak nie ma dwóch identycznych ludzi, nie ma też dwóch identycznych modeli. Nawet w przypadku bliźniaczych filii jednej firmy ich modele na pewno będą się różniły na poziomie pracowników.

Opisany wyżej model systemu informacyjnego zarządzania jednoznacznie określa strukturę logiczną Bazy Danych pakietu DIANA, w której będą przechowywane wszystkie wprowadzane dane. Jak trudne zadania stoją przed oprogramowaniem, nie tylko samej Bazy Danych, ale też całego bloku wprowadzania i aktualizacji danych, świadczy wymiarowość uzyskiwanej sieci – w niektórych obiektach sieć powiązań między zadaniami zawierała kilkadziesiąt tysięcy węzłów i kilkaset tysięcy łuków, zaś każdy z tych rekordów opisywany jest przez kilkanaście parametrów.



Rys. 18. Schemat funkcjonalny pakietu DIANA-11

Przy tak olbrzymiej ilości wprowadzanych danych musiał być przewidziany odpowiedni system bieżącej kontroli ich prawidłowości. Oczywiście w przypadku dużych obiektów nie do pomyslenia byłoby wprowadzenie wszystkich danych przez jedną osobę – trwałoby to zbyt długo, na pewno dłużej niż zmiany zachodzące w każdym „żywym” obiekcie. Dlatego już od pierwszej wersji przewidziano możliwość realizacji tego etapu równolegle przez dowolnie duży zespół. Mieliśmy więc następne wyzwanie – oprogramowanie musiało nie tylko „poskładać” w jedną spójną całość wysiłki poszczególnych członków zespołu, ale również pilnować dyscypliny w trakcie realizowanych przez nich prac, a w szczególności ich kolejności. Jest oczywiste, że brak będzie możliwości np. wskazania dostawców, jeżeli któryś z innych członków zespołu nie zdążył jeszcze z wykonaniem swojego odcinka pracy. Tak więc komputer musi pilnować, aby realizacja tego etapu była rzeczywiście pracą zespołową.

Na dalszych etapach (Rys.18.) uzyskany model może być wykorzystany do analizy diagnostycznej (Rozdział IV) i projektowania struktur organizacyjnych (Rozdział V) oraz Systemów Informowania

Kierownictwa (Rozdział VI), a następnie do reorganizacji, restrukturyzacji, reinżynierii, monitoringu i rozwoju organizacji (Rozdział VII).

Zanim jednak przystąpimy do tych etapów warto zająć się problemem bardziej ogólnym – możliwością uniwersalnego opisu badanych zależności.

III.5. Język opisu modelu.

Istnieje ścisła zależność między automatami, gramatykami i językami [20]. Mówiąc ogólnie, jeżeli automat traktować jako model, to odpowiadająca mu gramatyka stanowi zbiór reguł jego działania, a jednocześnie służy do definiowania języka formalnego, którego zadaniem jest generowanie programów realizujących model. Również w odwrotną stronę - automat rozpoznaje program napisany poprawnie, w przynależnym mu języku, zgodnie z odpowiednimi regułami gramatyki.

Spośród gramatyk interesować nas będą gramatyki kombinatoryczne.

Definicja 6.

Gramatyka kombinatoryczna jest o system

$$G = \langle I_k, I_p, P, \sigma \rangle \quad (120)$$

gdzie:

I_k - alfabet końcowy

I_p - alfabet pomocniczy

P - lista produkcji

$$P \subseteq \left((I_k \cup I_p)^* - \{\mathcal{E}\} \right) \times (I_k \cup I_k)^*$$

$\sigma \in I_p$ - symbol początkowy

Definicja 7.

Język definiowany przez daną gramatykę kombinatoryczną G jest to zbiór słów, który oznaczamy przez $L(G)$ i który składa się ze wszystkich słów nad alfabetem końcowym I_k wyprowadzalnych z σ w G

$$L(G) = \left\{ x : x \in I_k^* \wedge \sigma \xRightarrow[G]{*} x \right\} \quad (121)$$

I_p pozwala stworzyć pewnego rodzaju metajęzyk, w którym język L jest definiowany.

Gramatyki kombinatoryczne można podzielić na trzy grupy. Odpowiednio do tego można podzielić automaty i języki

- I. Gramatyki monotoniczne - automaty liniowo ograniczone - języki kontekstowe
- II. Gramatyki bezkontekstowe - automaty ze stosem - języki bezkontekstowe
- III. Gramatyki prawostronnie liniowe - automaty bez wyjścia - języki regularne

Wprowadzimy pojęcie automatu abstrakcyjnego (będzie ono uzupełnieniem definicji 5 - wzór (47)).

Definicja 8.

Automat abstrakcyjny jest to system

$$S^a = \langle A, I_D, I_O, F, R, A_0, \#_e, \#_p \rangle \quad (122)$$

gdzie:

- A - zbiór stanów
- I_D - alfabet wejściowy
- I_O - alfabet wyjściowy
- F - A - zbiór stanów końcowych
- R - zbiór instrukcji

$A_0 \in A$ - stan początkowy

$\#_e, \#_p \notin A, I_D, I_O$ - lewy i prawy wskaźnik końca słowa.

Można udowodnić równoważność automatu z klasy \mathcal{S} i gramatyk z klasy \mathcal{G} :

$$\bigwedge_{G \in \mathcal{G}} \bigvee_{S^a \in \mathcal{S}} S^a \Rightarrow G \wedge S^a \quad (123)$$

$$\bigwedge_{S^a \in \mathcal{S}} \bigvee_{G \in \mathcal{G}} G \Rightarrow A \wedge G \quad (124)$$

Z (123) i (124)

$$\bigwedge_{G \in \mathcal{G}} G \bigvee_{S^a \in \mathcal{S}} S^a \Rightarrow (G \wedge S^a) \text{ def } L \quad (125)$$

Interesować nas będzie II grupa - gramatyki bezkontekstowe. Zwróćmy uwagę, że I_k jest zbiorem elementów należących do badanego obiektu:

$A(i, j, r, s)$, $I_{Df}(i, j, r, s, f)$, $I_O(i, j, r, s)$ itd.,

zaś I_p - zbiór operatorów, np. \bigvee , \bigwedge , \sim itd.

Uwzględniając specyfikę naszego modelu spróbujemy określić poszczególne elementy (122).

A - odpowiada zbiorowi stanów wszystkich operacji, czyli (101)

I_D - odpowiada zbiorowi identyfikatorów wszystkich operacji dostawczych, czyli (99)

I_O - odpowiada zbiorowi identyfikatorów wyników wszystkich operacji, czyli (100)

$$F \equiv \bigwedge_{i,j,r,s=1}^{k,l,m,n} [A(i,j,r,s)=1] \quad (126)$$

$$A_O \equiv \bigwedge_{i,j,r,s=1}^{k,l,m,n} [A(i,j,r,s)=1] \quad (127)$$

R - zawiera wszystkie prawa i reguły klasycznego rachunku kwantyfikatorów oraz podstawową zależność opisującą funkcjonowanie modelu (119).

Odpowiadająca temu automatowi gramatyka (120) będzie definiowała język formalny, którym używając odpowiedniej symboliki:

$$\left(\mathbf{V}, \mathbf{\Lambda}, \sim, \vee, \wedge, \equiv, \Rightarrow, \text{identyfikatory, indeksy} \right),$$

posługiwaliśmy się dotychczas przy formułowaniu poprawnych zależności.

Opis tego języka najwygodniej jest przedstawić w notacji Backus'a. Niżej podany jest przykładowo zapis fragmentu proponowanego języka:

<zadanie>	::	=	<wskaźnik> <słowo> <wskaźnik><słowo> $\cdots L$
<wskaźnik>	::	=	spacja (odpowiada $\#_1, \#_p$)
<słowo>	::	=	<identyfikator> <wartość logiczna> <ogranicznik>
<identyfikator>	::	=	<litera> <cyfra>
<litera>	::	=	duże i małe litery alfabetu łacińskiego
<cyfra>	::	=	cyfry arabskie
<wartość logiczna>	::	=	$P F$ (odpowiada logicznemu 1, 0)
<ogranicznik>	::	=	<operator> <przerywnik> <nawias> <miano> <specyfikator>
<operator>	::	=	<kwantyfikator ogólny> <kwantyfikator egzystencjalny> <operator logiczny> <operator relacji> <operator arytmetyczny> <operator następstwa>

<kwantyfikator ogólny>	:: = Λ
<kwantyfikator egzystencjalny>	:: = $V V_1$
<operator logiczny>	:: = $\vee \wedge \sim \equiv \Rightarrow$
<operator relacji>	:: = $< \leq = > \geq \neq$
<operator arytmetyczny>	:: = $+ - \times / \div \uparrow$
	...

Z przyczyn zrozumiałych zrezygnowano z opracowania języka „od podstaw”. Natomiast uzyskany metajęzyk w znacznym stopniu ułatwił opracowanie spójnego opisu modelu w postaci użytkowej. Nader inspirującą w tym zakresie była praca [102]. Otrzymana postać użytkowa opisu modelu [79] pozwoliła na bezpośrednie tworzenie schematu Bazy Danych i generowanie Bazy Danych za pomocą oprogramowania MIDAS.

Widzimy, że tworzenie zdań języka opisu danych jest w tym przypadku bardzo proste. Metajęzyk był również nader użyteczny przy tworzeniu generatora programów w pakiecie DIANA-9 oraz programu RAPORT.

XI. BIBLIOGRAFIA

- [1] "ADW. Technical Reference, Knowledge Ware", London 1990.
- [2] "Algebraic theory of machines, languages and semigroups". Edited by M. A. Arbib; Academic Press, N-Y, London 1963
- [3] "Analyst Workbench", Infotech State of the Art Report, Maidenhead, 1987.
- [4] M. C. Barnes, A. M. Fogg, C. N. Stephens, L. G. Fitman : "Organizacja przedsiębiorstwa . Teoria-praktyka", PWE, Warszawa, 1972.
- [5] A. Barski, E. Michalewski : "DIANA-9. Pakiet wspomaganiej komputerowo analizy diagnostycznej i projektowania systemów zarządzania", Akademickie Forum Informatyki - INFOSYSTEM'94, Poznań 1994r.
- [6] A. Barski, E. Michalewski : "Komputerowa diagnostyka dużych sieci informacyjnych" DPP'2001, Łagów, 2001.
- [7] A. Barski, E. Michalewski : "Komputerowe monitorowanie zagrożeń organizacyjnych" Materiały konferencji KSW'2001 (Ciechocinek 5 – 7 września 2001 r.)
- [8] A. Barski, E. Michalewski : "Komputerowe wspomaganie procesu wdrażania dużych systemów informatycznych", BIS'99 - Poznań, kwiecień 1999r.
- [9] A. Barski, E. Michalewski : "Metodyka DIANA, a narzędzia klasy Workflow", w : "Komputerowe wspomaganie Zarządzania i Procesów Decyzyjnych w Gospodarce", Wyd. IBS PAN, Warszawa 2002
- [10] A. Barski, E. Michalewski : "Pakiet DIANA-9 (opis funkcjonowania pakietu)", Konf. : "Informatyka na wyższych uczelniach dla gospodarki narodowej" (Tempus Joint European Project), Gdańsk 1994
- [11] A. Barski, E. Michalewski : "Wspomagana komputerowo analiza diagnostyczna i projektowanie systemów zarządzania - pakiet DIANA-10", Wykład i demonstracja pakietu w ramach "Tutorials" na II Międzynarodowej Konferencji "Business Information Systems" BIS'98, Poznań, kwiecień 1998r.
- [12] A. Barski, E. Michalewski : "Wykorzystanie metodyki DIANA w Stoczni Gdynia S. A.", w : "Społeczeństwo informacyjne a badania operacyjne i zarządzanie", Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002

- [13] A. Barski, E. Michalewski, H. Niedźwiedzińska, I. Rakhmanova, M. Pashkin, A. Smirnov : "Analiza porównawcza ocen grupowych ekspertów dotyczących przydatności czynników wpływających na decyzję o wdrożeniu" EDI, VI Międzynarodowa Konferencja EDI'98, Łódź-Dobieszków, maj 1998 (wydrukowane : wyd. Uniwersytet w Łodzi).
- [14] A. Barski, E. Michalewski, M. Pashkin, I. Rakhmanova, A. Smirnov : "Application of Decision Support Tools in Organization Management" Systems Sciences'2001, Wrocław, 2001.
- [15] A. Barski, E. Michalewski, M. Pashkin, I. Rakhmanova, A. Smirnov : "Concepts, methods and tools of business process computer-aided reengineering", "The Fifth International Conference on Advanced Computer Systems" ACS'98, Szczecin, listopad 1998r.
- [16] A. Barski, E. Michalewski, M. Pashkin, I. Rakhmanova, A. Smirnov : "Configuration management for business process reengineering : : concepts, methods and special tools". AMETMAS'99, St. Petersburg, 1999.
- [17] A. Barski, E. Michalewski, M. Pashkin, I. Rakhmanova, A. Smirnov : "Zintegrowane środowisko wspomaganie komputerowo reinyżierii dużych przedsiębiorstw", Konf. Naukowa "Badania Operacyjne i Systemowe" BOS'98, czerwiec 1998.
- [18] A. Barski, E. Michalewski, I. Rakhmanova, A. Smirnov : "Organization Management Decision Support Tools For Manufacturing Systems Re-Engineering", III Internat. Conf. "Information Development System" (IDS'97) St. Petersburg, czerwiec 1997r.
- [19] Z. Biniek : "Systemowo-diagnostyczna strategia projektowania informatycznego systemów zarządzania" W : "Problemy projektowania systemów informatycznych zarządzania", Politechnika Szczecińska, Szczecin, 1979.
- [20] A. Blikle : "Automaty i gramatyki", PWN, Warszawa 1971
- [21] W. A. Bocchino : "Systemy informacyjne zarządzania. Narzędzia i metody.", WNT, Warszawa, 1975
- [22] S. Brinkkemper, S. Hong, A. Bulhuis, G. van den Goor : "Object-Oriented Analysis and Design Methods a Comparative" Review, University of Twente, 1998 ([http : //elex. amu. edu. pl/languages/oodoc/oo-a. html](http://elex.amu.edu.pl/languages/oodoc/oo-a.html) – styczeń 2003 r.)
- [23] W. Chmielarz : "Ocena systemów informatycznych dla małych i średnich firm – aspekt modelowy"; w : "Komputerowe wspomaganie Zarządzania i Procesów Decyzyjnych w Gospodarce", Wyd. IBS PAN, Warszawa 2002
- [24] "Current Trends in Information Systems Development Methodologies",

- Preprints of the Polish-Scandinavian Seminar Paraszyno, June 1988.
- [25] M. Dolińska : "Modelowanie zintegrowanego systemu informacyjnego przedsiębiorstwa"; Informatyka 7-8/99, wyd. Sigma, Warszawa 1999
- [26] L. Drelichowski : "Zastosowanie metod optymalizacyjnych w systemach logistyki jako pochodne zmian organizacyjnych i softwerowych"; w : "Społeczeństwo informacyjne a badania operacyjne i zarządzanie", Akademicka Oficyna Wydawnicza EXIT, Warszawa 2002
- [27] P. Drożdżyk, R. Krutak, R. Markiewicz, J. Ostrowski : "Mikrokomputerowe wspomaganie procesów projektowania usprawnień organizacyjnych w systemach informacyjno - decyzyjnych", Krajowa Konferencja "Komputerowe systemy i metody wspomagające podejmowanie decyzji". Warszawa 1987.
- [28] P. F. Drucker : "The Practice of Management", London, 1958.
- [29] P. H. Duffin : "Knowledge based systems - applications in administrative government", Ellis Horwood Ltd, Chichester, 1989.
- [30] I. Durlik : "Restrukturyzacja procesów gospodarczych. Reengineering, teoria i praktyka", Placet, Warszawa 1998.
- [31] A. Dzianott : "Podstawy metodologii i projektowania systemów informatycznych wg metody 'MERISE'. Ogólnopolska konferencja - Techniki Komputerowe w Zarządzaniu Produkcją" INFOPROD'91. Bydgoszcz 1991.
- [32] A. Elek, T. Rawiński, S. Wrycza : "Charakterystyka wybranych narzędzi komputerowego wspomagania tworzenia systemów informatycznych", Prace badawcze Politechniki Gdańskiej, nr 162, 1989.
- [33] W. L. Epsztejn, W. I. Seniczkin : "Językowe środki architekta ASU", "Energia", Moskwa 1979.
- [34] R. Gabryelczyk, M. Lasek : "Modelowanie procesów gospodarczych za pomocą ARIS-TOOLSET", UW, Warszawa 1998.
- [35] Z. Gackowski : "Projektowanie systemów informacyjnych zarządzania", WNT, Warszawa, 1974.
- [36] M. L. Gibson : "The CASE Philosophy", BYTE, April 1980, pp. 209-218.
- [37] R. W. Griffin : "Podstawy zarządzania organizacjami", PWN, Warszawa 1998
- [38] W. M. Grudzewski, I. K. Hejduk : "Koncepcja kreowania organizacji inteligentnej w przedsiębiorstwach", Organizacja i Kierowanie, nr 4, 1997.
- [39] W. Grudzewski, I. Hejduk : "Projektowanie systemów zarządzania", Difin, 2000.

- [40] W. Grudzewski, I. Hejduk : "Przedsiębiorstwo przyszłości", wyd. Difin, Warszawa 2000
- [41] W. Grudzewski, I. Hejduk : "Przemiany w technice i technologii prognozy XXI wieku", Wyd. Ekonomia i Organizacja Przedsiębiorstw, nr 11/98, Warszawa 1998
- [42] A. Grzegorzczak : "Zarys logiki matematycznej", PWN, Warszawa 1979
- [43] J. R. Hackman, G. R. Oldham : "Motivation Through the Design of Work", N-Y, 1976
- [44] M. Hammer, J. Champy : "Reengineering w przedsiębiorstwie", Neumann Management Institute, Warszawa 1996.
- [45] J. E. E. Hijmans : "Pratique de l'organisation industrielle", Paris, 1954.
- [46] "HIPO : Documentation Structure Design", Auerbach Publishers Inc. Philadelphia 1979.
- [47] <http://www.micrografx.top.pl>
- [48] R. Keller : "Expert System Technology (Development and Application)", Prentice-Hall Company, Englewood Cliffs, New Jersey 1987.
- [49] W. Kieżun : "Sprawne zarządzanie organizacją", wyd. SGH, Warszawa 1997
- [50] J. Kisielnicki, H. Sroka : "Systemy informacyjne biznesu", Agencja Wydawnicza Placet, Warszawa, 2001.
- [51] T. Kotarbiński : "Traktat o dobrej robocie", Z-d im. Ossolińskich, Wrocław 1975
- [52] M. Lundeberg : "The ISAC Approach to Specification of Information Systems and its Application to the Organization". IFIP Working Conference, North-Holland, Amsterdam, 1988.
- [53] J. Martin, C. McClure : "Structured techniques Basis for CASE", Prentice Hall, New York 1988.
- [54] V. Merlyn, G. Boone : "CASE Product Classification Model", CASE Bulletin, March 1989.
- [55] "Meta Edit+, Dokumentacja firmowa", 2002 r.
- [56] "META-SIKOP. Raport prac wykonanych w 1975 - 1979"; ORGMASZ Warszawa 1979.
- [57] E. Michalewski : Computer - "Aided Advisor for organization management based on the package DIANA - 9". Int. Sem. "Operational and Systems Research of the Transition to Advanced Market Economies", Bratislava 1990.
- [58] E. Michalewski : "Algorithm for automatization of the first step of design of organization structure with the use of DIANA-9 package"; 9-th Polish-Italian & 6-th Polish-Finnish Conf. "Systems analysis and

- Decision Support in Economics and Technology” Radziejowice (Poland) 1993.
- [59] E. Michalewski : ”Application of a microcomputer package DIANA-8 for design computerized management systems”; II Polish - Scandinavian Seminar ”Current trends in information systems development methodologies”, Gdańsk 1990.
- [60] E. Michalewski : ”Computer-Aided Design Executive Information Systems”, III Internat. Conf. ”Information Development System” (IDS'95) St. Petersburg 1995r.
- [61] E. Michalewski : ”Computer-aided diagnostic analysis and design of information systems implemented on PC as a package DIANA-9”, Gdańsk 1992
- [62] E. Michalewski : ”DIANA-9 - pakiet wspomaganego komputerowo analizy diagnostycznej i projektowania struktur organizacyjnych”; ”Informatyka” Nr 11, 1992.
- [63] E. Michalewski : ”Formalizacja wybranych funkcji systemu zarządzania jednostką gospodarczą”. W : ”Metody cybernetyczne w zarządzaniu”, Warszawa 1974. Wyd. Ossolineum, Wrocław 1978
- [64] E. Michalewski : ”Komputerowo wspomagany system zarządzania Stoczną Gdynia S. A.”, KSW 2000, Ciechocinek 2000, wyd. IBS PAN, Warszawa 2000.
- [65] E. Michalewski : ”Mikrokomputerowa baza danych dla potrzeb symulacji dużych sieci”, IV Ogólnopolskie Sympozjum SPD-4 ”Symulacja procesów dynamicznych”. Zakopane 1987.
- [66] E. Michalewski : ”Mikrokomputerowy pakiet wspomaganego analizy diagnostycznej i projektowania struktur organizacyjnych”; IV Górska Szkoła Informatyczna, 1992.
- [67] E. Michalewski : ”Modern methods of computer-aided analysis and design of management systems”; Milano 1992.
- [68] E. Michalewski : ”Multilevel polyhierarchical model for organizational decision support implemented on IBM PC type package DIANA-9”; International Conf. ”Support Systems for Decision and Negotiation Processes”, Warszawa 1992.
- [69] E. Michalewski : ”Nowy trend w CAMS - komputerowy lekarz systemu zarządzania”; INFOGRYF 90, Szczecin 1990.
- [70] E. Michalewski : ”Package for computer-aided diagnostic analysis and design of management systems”; Intern. Workshop ”Intelligent Decision Support Systems” IDSS'92, Kuzively (Crimea) Ukraine 1992
- [71] E. Michalewski : ”Pakiety DIANA-10 jako platforma integrująca specjalistów różnych dziedzin”, Konf. Naukowa ”Badania Operacyjne

- i Systemowe” BOS'95, Szczecin 1995r.
- [72] E. Michalewski : ”Polyhierarchical dynamic model of a large - scale management system”; Prace Naukowe ICT PW, Nr 3, Wrocław 1978.
- [73] E. Michalewski : ”Problemy przeniesienia pakietu symulacji dużych sieci informacyjnych na technikę mikrokomputerową”, III Ogólnopolskie Sympozjum SPD-3 ”Symulacja procesów dynamicznych” Zakopane 1986.
- [74] E. Michalewski : ”Projektowanie systemów zasilających decydentów w informacje o najwyższym priorytecie dla decydentów”, XII Kołobrzeskie dni informatyki INFOGRYF'94 Kołobrzeg 1994r.
- [75] E. Michalewski : ”Reorganizacja, restrukturyzacja, re-engineering?”, III Konferencja ”Komputerowe systemy wielodostępne”, Bydgoszcz-Ciechocinek, wrzesień 1997r.
- [76] E. Michalewski : ”Some aspects of computer diagnostic analysis of the management systems”; ”Control and Cybernetics”, vol. 4 No 3 - 4, 1975.
- [77] E. Michalewski : ”Tworzenie środowiska przyjaznego dla EDI”, III Kraj. Konf, EDI, Łódź 1995r.
- [78] E. Michalewski : ”Wersja edukacyjna pakietu DIANA-9 - wspomaganej komputerowo analizy diagnostycznej i projektowania systemów zarządzania”, Konf. : ”Informatyka na wyższych uczelniach dla gospodarki narodowej” (Tempus Joint European Project), Gdańsk 1994r.
- [79] E. Michalewski : ”Wieloprocessorowy model dynamiczny dużych sieci”; V Ogólnopolskie Sympozjum SPD-5 ”Symulacja procesów dynamicznych”, Zakopane 1988.
- [80] E. Michalewski : ”Wskaźniki rozmyte przy projektowaniu dużych sieci”; VI Ogólnopolski Sympozjum SPD-6 ”Symulacja procesów dynamicznych”, Zakopane 1990.
- [81] E. Michalewski : ”Wspomagane komputerowo diagnoza i projektowanie systemów informacyjnych zarządzania”, wyd. WSISiZ, Warszawa 2003.
- [82] E. Michalewski : ”Wspomagane komputerowo projektowanie nowych organizacji”; 3 Konferencja Badań Operacyjnych i Systemowych BOS'93, Warszawa 1993.
- [83] E. Michalewski : ”Wspomagane komputerowo projektowanie systemu wczesnego ostrzegania dla banku centralnego”, Kraj. Konf. : ”Analiza decyzyjna, systemy eksperckie, zastosowania systemów komputerowych”, Warszawa 1994r.
- [84] E. Michalewski : ”Wykorzystanie metodyki DIANA w procesie projektowania i wdrażania” ISWD, Konferencja naukowa

- "Inteligentne Systemy Wspomagania Decyzji w Zarządzaniu", Katowice-Wisła, październik 1997r.
- [85] E. Michalewski : "Wykorzystanie pakietu DIANA-10 w przygotowaniu przedsiębiorstwa do wdrożenia EDI", IV Międzynarodowa Konferencja EDI'96, Łódź-Arturówek, maj 1996 (wydrukowane : wyd. Uniwersytet w Łodzi).
- [86] E. Michalewski : "Wykorzystanie pakietu DIANA-9 w procesie restrukturyzacji przedsiębiorstw", Międzynarodowa Konferencja "Business Information Systems '97" (BIS'97), Poznań, kwiecień 1997.
- [87] E. Michalewski : "Wykorzystanie techniki mikrokomputerowej do projektowania systemów informatycznych"; Międzynarodowa konferencja "Nowoczesne metody zarządzania", Wrocław 1990.
- [88] E. Michalewski : "Zastosowanie Q-algebry do komputerowego projektowania dużych sieci"; Zeszyty Naukowe WSI, ser. "Elektryka" z. 15, Opole 1980.
- [89] E. Michalewski, R. Markiewicz, J. Ostrowski : "Pakiet DIANA-8 do wspomagania decyzji organizatorskich w sferze zarządzania przedsiębiorstwa" : INFOGRYF'88, Kołobrzeg 1988
- [90] E. Michalewski, H. Niedźwiedzińska : "Komputerowo wspomagane diagnozowanie potrzeb organizacji w zakresie elektronicznej wymiany danych", V Międzynarodowa Konferencja EDI'97, Łódź-Dobieszków, czerwiec 1997 (wydrukowane : wyd. Uniwersytet w Łodzi).
- [91] E. Michalewski, J. Ostrowski : "Komputerowy model sfery zarządzania przedsiębiorstwa do wspomagania analiz systemowych". Międzynarodowa Konferencja "Badania Operacyjne i Systemowe" BOS'88, Książ k/Wałbrzycha 1988
- [92] E. Michalewski, J. Ostrowski : "Practical questions of applying computers to analysis and design of management systems". MECO'83, Ateny 1983.
- [93] E. Michalewski, J. Ostrowski, M. Stankiewicz : "Computer-aided diagnosis and design of plant organization"; AMPS - COMPCONTROL'85, Budapeszt 1985.
- [94] E. Michalewski, J. Ostrowski, M. Stankiewicz : "Pakiet DIANA-6 jako narzędzie do modelowania, analizy i projektowania systemu sterowania przedsiębiorstwem". III Konferencja "Zastosowanie komputerów w przemyśle". Szczecin 1983.
- [95] E. Michalewski, J. Ostrowski, M. Stankiewicz : "The concept of a software tool for analysis and simulation of decision and information flow in large-scale organization"; The First IASTED Symposium,

- Lille 1983.
- [96] J. Nadler : "Design information systems. Practical approach", McGraw Hill Publ., London 1987.
- [97] E. Niedzielska : "Projektowanie systemów informatycznych". PWE, Warszawa 1977
- [98] J. Nowicki : "Modernizacja systemu informacyjnego w przedsiębiorstwie przemysłowym". PWE. Warszawa, 1979.
- [99] J. Nunamaker : "A Methodology for the Design and Optimization of Information Processing Systems". AFIPS Conference Proceedings, Vol. 38, 1971.
- [100] J. Ostrowski : "Group Assignment problem", AMPS`85, Budapest, 1985.
- [101] "PACBASE (trade information); CGI Systems Inc"; N-Y, 1990.
- [102] S. Piasecki : "Teoria organizacji w świetle analizy systemowej jako teoria języka problemowo zorientowanego", Prace IBS PAN, Vol. 82, 83, Warszawa 1982
- [103] "Podstawy ekonomii", red. Milewski R., PWN, Warszawa 2003
- [104] A. W. Pogorzelski : "Klasyczny rachunek zdań", PWN, Warszawa 1973
- [105] W. V. Quine : "Logika matematyczna"; PWN, Warszawa 1974
- [106] S. E. Savory : "Expert systems in the organization (an introduction for decision- makers)" John Wiley & Sons, N-Y 1988.
- [107] A. W. Scheer : "Business Process Engineering. Reference Models for Industrial Enterprises", Springer-Verlag, 1994.
- [108] A. W. Scheer, C. Kocian, U. Markus : "Od modelowania danych do modelowania wiedzy – struktury, narzędzia"; Informatyka 2/98, wyd. Sigma, Warszawa 1998
- [109] J. A. F. Stoner, Ch. Wankel : "Kierowanie", PWE Warszawa 1997
- [110] STRATEGOR : "Zarządzanie firmą", PWE, Warszawa 1996
- [111] "System Development Workbench, CGP Case-Tools", Rijswijk 1991.
- [112] J. Szczupaczyński : "Anatomia zarządzania organizacją" MSM, Warszawa, 1998.
- [113] Teichrow : "PSL/PSA - Technical Reference", MIT Rep., 1980.
- [114] Teichrow, Gackowski : "Comparison Analysis of Methods for Design Information Systems", MIT Reports, Masuchet 1979.
- [115] J. Trzcieniecki, A. Stabryła : "Zagadnienia metodologii badania systemów zarządzania", AE, Kraków, 1980.
- [116] J. D. Warnier : "New method of design information flow in large systems", Honeywell-Bull Rep., Toulouse 1974.
- [117] S. Wrycza : "Aktualne trendy komputerowo wspomaganego tworzenia systemów informatycznych", Trzecia Wiosenna Szkoła PTI,

Swinoujście 1990.

- [118] S. Wrycza : "Współczesne metodyki tworzenia systemów informatycznych zarządzania". PTC, Gdańsk 1989.
- [119] "Zarządzanie (teoria i praktyka)", red. Koźmiński A., Piotrowski W., PWN, Warszawa 1997
- [120] J. Zieleniewski : "Organizacja i zarządzanie", PWN, Warszawa, 1979.

