STRUKTURALNE PROJEKTOWANIE NA TLE INNYCH METOD PROJEKTOWANIA

"Informatyka przypomina w pewnym sensie energię jądrową; jest to jedna z tych rzeczy, które niesłychanie długo nadchodzą i mają zwodniczy charakter, bo wydają się być w zasięgu ręki, ale jakoś nigdy nie dają się uchwycić."

/P.Hermon, British Airways/

Jeśli komputer można uznać za maszynę parową XX wieku, o tyle informatyka - jako dziedzina wiedzy - przypomina swoisty labirynt, w który maszyna ta podąża do celu, tracąc wiele swojej energii. Chodzi o to, iż mimo wysokiej sprawności technicznej komputerów, systemy informatyczne wspomagające zarządzanie dostarczają informacji wynikowych ze znacznym opóźnieniem, zaś budowa ich trwa lata. Będąc dziedziną interdyscyplinarną, informatyka korzysta w sposób naturalny z dorobku innych dziedzin wiedzy, równocześnie sama umożliwiając dalszy ich rozwój. Odnosi się to szczególnie do projektowania strukturalnego, które jest działem informatyki podającym reguły inżynierii projektowania systemów informatycznych i ściśle związanym z ogólnymi metodami projektowania.

W rozdziale tym nie podejmujemy się dokonywania przeglądu i oceny tych metod, lecz zamierzamy zasygnalizować niektóre odrębności i powinowactwa w stosunku do metody projektowania strukturalnego. Pozwoli to lepiej zrozumieć jej genezę oraz przewidzieć dalszy rozwój.

Naszym zdaniem najbardziej kontrowersyjne powiązania dotyczą teorii /techniki/ systemów. Ogólnie rzecz biorąc, w podejściu systemowym przyjmuje się, że system jest to zestaw powiązanych składników przeznaczony do realizacji określonego celu, przy czym przez zestaw składników rozumie się elementy składowe wraz z relacjami pomiędzy nimi oraz relacjami w stosunku do całości. Upod-

staw tej definicji przyjęto pogląd Halla A.D. /19, s.93/, że "system jest to zbiór obiektów wraz z relacjami istniejącymi pomiędzy tymi obiektami oraz pomiędzy ich własnościami" oraz twierdzenie autorów pracy /23, s.113/, którzy określili system jako "zestaw komponentów zaprojektowany do realizacji wytyczonych celów zgodnie z planem".

Projektowanie strukturalne przejmuje z projektowania systemowego zasadę wzajemnego powiązania składników jako oczywisty warunek spójnej struktury systemu informatycznego. Natomiast samo pojęcie "system" nie kojarzy się tutaj z "całością" jako czymś funkcjonalnie i strukturalnie zamkniętym. To "odchylenie" tłumaczy się tym, że w odróżnieniu od systemu materialnego "system" informatyczny nigdy nie jest kompletny /całkowity/ i w związku z tym relacje składników z całością noszą charakter cząstkowy. Niekompletność systemu informatycznego polega na tym, że nie stanowi on odwzorowania systemu rzeczywistego np. przedsiębiorstwa /jako całości/, gdyż realizowany jest fragmentarycznie i w ciągu tak długiego okresu czasu, że nie uwzględnia dostatecznie /w porę/ zmian zachodzących w tym systemie. Ponadto system informatyczny jest tworem niedoskonałym, którego elementy ciągle /praktycznie/ podlegają korektom i modyfikacjom.

Przedstawiciele teorii systemów podkreślają znaczenie "całości" dla funkcjonowania systemu. Na przykład B. Hart sformułował następujące zasady spójnego działania systemu /20, s.125/:

- całość jest najważniejsza, zaś części zawsze mają znaczenie drugoplanowe i wynikają /wraz ze swymi powiązaniami/ z koncepcji całości,
- cel postawiony przed całością determinuje rolę składników,
- to, co części realizują i jak to robią, całkowicie zależne jest, zarówno od ich pozycji w całości, jak i powiązań, jakie istnieją między całością a składnikami, z których jest zbudowana.

Z dominującej roli całości wynika również stabilna struktura systemu teorio-systemowego. W ogólnej teorii systemów /28, s.210 -215/ występuje struktura systemu jako "ta część organizacji, która jest trwała, ustalona lub niezmienna i tworzy podstawę dla zachowania względnie stałego lub stałego." Przez organizację

systemu rozumie się tutaj zespół właściwości, których wynikiem jest zachowanie się systemu.

Jeśli przyjmiemy, że system informatyczny nie stanowi odwzorowania systemu przedmiotowego, z którego pochodzą dane wejściowe, wówczas również wypada dopuścić twierdzenie, że struktura systemu informatycznego nie musi wynikać ze struktury problemów, które obsługuje, lecz można ją tworzyć według swoistych kryteriów technologicznych. Ocena jakości tych kryteriów powinna być odzwierciedleniem stopnia gotowości systemu informatycznego do rozwiązywania problemów nieoczekiwanych, nie przewidzianych w gotowych programach. Innymi słowami, chodzi tutaj o zdolność do rozwoju i modyfikacji. Można ją osiągnąć poprzez sprawne mechanizmy sprzęgające, u podstaw których leżeć mogą następujące relacje specyfikowane w metodologii badań systemowych /63,s.36-37/:

- sprzężenia współdziałania /wynikające z podobieństwa celów/,
- sprzężenia genetyczne /występujące w sytuacji, gdy jeden element systemu stanowi podstawę utworzenia elementu innego/,
- sprzężenia przekształcenia /charakteryzujące przechodzenie obiektów z jednego stanu w drugi/,
- sprzężenia strukturalne /realizujące określony typ struktury, np. hierarchicznej lub sieciowej/,
- sprzężenia funkcjonalne /wynikające ze wspólnego wykonywania określonej funkcji/,
- sprzężenia rozwojowe /wynikające nie tyle ze zmiany stanów podczas realizacji funkcji, ile z takiej zmiany stanów, że towarzyszą jej istotne zmiany strukturalne i zmiany zachowania się obiektu,
- sprzężenia sterowania /niekiedy stanowiące rodzaj sprzężeń funkcjonalnych lub rozwojowych/.

Liczba sprzężeń w systemie informatycznym może być bardzo znaczna. W warunkach swobodnych dwukierunkowych sprzężeń sieciowych /każdego elementu z każdym/ dla zbiorowości liczącej 1 000 elementów sięga ona prawie dwa miliony. Zważywszy na fakt, że każde sprzężenie realizowane jest albo poprzez odsyłacze adresowe albo instrukcje wywołujące, realizacja tych możliwości odbywa się kosztem znacznych nakładów pamięciowych i pochłania sporo

mocy obliczeniowej komputerów. W praktyce przeciwdziała się wzrostowi liczby relacji, stosując struktury hierarchiczne lub ograniczone struktury sieciowe /np. typu set według propozycji organizacji CODASYL/. Nie zabezpiecza to jednak przed koniecznością reorganizacji baz danych w przypadku pojawienia się innych relacji niż te, które w schemacie bazy przewidział projektant, przy wyczerpaniu ewentualnych pól rezerwowych na dodatkowe relacje.

Gdyby system informatyczny był oparty o model matematyczny systemu przedmiotowego /materialnego/, czyli posiadałby strukturę spełniającą w dostatecznym stopniu warunki podobieństwa do struktury modelowanego systemu, wówczas możliwe byłoby określenie już w momencie generowania baz danych wszystkich efektywnych relacji pomiędzy zmiennymi. Ponieważ taka idealna sytuacja nie istnieje, przed projektowaniem strukturalnym stawia się trudne zadanie skonstruowania zmiennych /pływających/ relacji występujących zarówno pomiędzy danymi, jak i pomiędzy procedurami oraz procedurami i danymi.

Jeśli, idąc dalszym tokiem naszych rozważań, zgodzimy się, że system informatyczny na obecnym etapie rozwoju informatyki nie stanowi ani modelowego, ani strukturalnego odwzorowania systemu przedmiotowego, a mimo wszystko "w jakiś sposób" ma zaspokajać potrzeby informacyjne tego systemu, to dojść możemy do wniosku że posiada on swoje specyficzne mechanizmy samoorganizacji pozwalające mu adaptować się do zmiennych warunków środowiska.

W tym momencie odchodzimy więc od preferującego stabilną strukturę podejścia systemowego do ujęcia cybernetycznego. Jeśli w teorii systemów system możemy zdefiniować jako strukturę zdolną do celowych działań, wynikających z właściwości i celów całości, to w metodzie projektowania strukturalnego system jest zmienną strukturą zdolną do działań wynikających z potrzeb danego problemu, danej sytuacji czy też indywidualnych potrzeb poszczególnych użytkowników. Takie potraktowanie systemu informatycznego wpływa radykalnie na następujące elementy procesu budowy systemu:

- podmiot /kwalifikacje projektantów, programistów, technologów/,
- tworzywo /zestawy danych, procedury/,

- środki /techniki, narzędzia/,
- metody,
- cele,
- wytwory /struktury użytkowe obsługujące użytkowników/.

Przedmiotem naszego zainteresowania w niniejszym rozdziale są metody projektowania, przez które rozumiemy całokształt ogólnych założeń i wytycznych w postępowaniu projektowym, oparty o określone zasady projektowe.[1/] W prakseologii metodą jest sposób umyślnie i systematycznie stosowany /29, s.708/,co interpretowane jest jako "sposób wykonywania czynu złożonego, polegający na określonym doborze i układzie jego działań składowych, a przy tym uplanowiony i nadający się do wielokrotnego stosowania" /30, s.88/. W metodzie wyróżnić można szereg etapów, których realizacja wymaga stosowania różnych technik i narzędzi.

U źródeł metody leżą zasady metodyczne, często inspirowane przez rozmaite tendencje filozoficzne i badawcze. Z punktu widzenia strukturalnego projektowania interesująco wyglądają inspiracje pochodzące z teorii zdarzeń, teorii przypadkowości /contingency theory/ i teorii sytuacyjnej /situational theory/ oraz takich metod projektowania systemów informacyjnych jak metoda współuczestnictwa /participative approach/, metoda socjo-techniczna /socio-technical approach/ i metoda infologiczna /infological approach/. W sumie wydają się one zabezpieczać odpowiednie łącze systemu informatycznego ze środowiskiem które ono obsługuje.Jednakże w żadnym stopniu nie rozwiązują one sprawy wewnętrznej konstrukcji systemu informatycznego.

Teoria zdarzeń jest metodologią ogólną teorii systemów działania, czyli szerszą dyscypliną niż teoria działania i prakseologia. Na gruncie tej teorii system przedmiotowy /np. przedsiębiorstwo przemysłowe/ może być traktowany jako system działań /np.przetwórczych/, zaś system informatyczny jako systemem obsługi transakcji.Według teorii zdarzeń /14, s.164/ system działania jest to uporządkowany zbiór układów, pomiędzy którymi występują

1/ Znane są np.takie zasady projektowania jak:zasada jedności funkcji, konstrukcji i formy, zasada równomiernego zużycia składników systemu,zasada optymalnego tworzywa,itp. /3, s.301/.

stosunki, przy czym za układ uważa się tutaj wszelkie stosunki uporządkowania elementów w pewne całości lub w zbiorze bez konieczności zachowania stosunków zależności wzajemnej /przykładem układu jest układ pierwiastków chemicznych Mendelejewa/. Zdarzenia są działaniami, które wywołują stan obiektu lub powodują, jako procesy, zmianę stanów.

W tym miejscu teoria zdarzeń zgodna jest z filozoficzną teorią A.N. Whiteheada /58, t.3 s.444/, który twierdził, że w przyrodzie występują dwa rodzaje elementów: zdarzenia - jako wytwory krótkotrwałe zwane również aktami, oraz obiekty /substancje/ - jako składniki stałe istniejące niezależnie od zdarzeń zwane w teorii bytów bytami.

Ch.B. Broad, przedstawiciel brytyjskiego nurtu filozoficznego zwanego realizmem analitycznym, określa zdarzeniem to wszystko, co dłużej lub krócej trwa /58, t.3 s.318/.

T. Kotarbiński /29. s.708/ twierdzi, że wszelkie zdarzenie ma materiał, czyli to z czym się coś dzieje, jakość, czyli ustosunkowanie stanu końcowego do początkowego i tok, inaczej przebieg. Przebiegu nie należy utożsamiać z procesem, gdyż proces występuje jeśli zdarzenie jest zmianą o określonej kierunkowości /np. proces wzrostu/.

O. Johnson /24, s.643/ traktuje zdarzenie jako "działanie, pojawienie się lub wydarzenie, które może być opisane przez niezliczoną ilość właściwości, atrybutów i charakterystyk".

W metodzie projektowania strukturalnego zdarzenie może być traktowane jako elementarna jednostka wejściowa do systemu, przy czym pozostałymi elementami systemu są obiekty i procesy. Przez termin "proces" rozumiemy tutaj ciąg zdarzeń powiązanych celem /jest to więc interpretacja nieco inna niż w teorii zdarzeń/. Przykładowo, proces zaopatrzenia obejmuje takie zdarzenia jak plan produkcji wyrobu, plan zużycia materiałów i zaopatrzenia, zamówienie na materiały, dostawa materiału, zapłata. Poza zdarzeniami wyróżnia się niekiedy transakcje do oznaczania zespołu zdarzeń, co jest zgodne z konwencją przyjętą w rachunkowości /np. 37, s.36/. Na przykład, transakcja kupna-sprzedaży składa się ze zdarzenia "dostawa" i zdarzenia "zapłata".

Informacyjnym odbiciem zdarzenia gospodarczego jest zdarzenie informacyjne, przy czym nie musi być to odwzorowanie wierne. Zdarzenia informacyjne wchodzą w relacje z wieloma logicznymi zestawami danych, służącymi do odwzorowań procesów, powiązań pomiędzy obiektami, itp. Podstawą relacji są atrybuty identyfikacyjne /klucze/ i atrybuty informacyjne. Atrybuty identyfikacyjne służą głównie do rozpoznania obiektów, są nimi np.symbole indeksowe materiałów w zdarzeniu dostawy.Atrybutami informacyjnymi mogą być: właściwości obiektu, cel zdarzenia, przyczyna, czas, miejsce, sprawca,narzędzie za pomocą którego dokonano zdarzenie /np.środek transportowy/, skutek zdarzenia /np.ilość materiału/. Kwalifikacja obiektów i atrybutów jest zmienna, zależnie od powiązań. Ten sam element może być w jednym zdarzeniu atrybutem /np.sprawcą w zamówieniu/, zaś w innym - obiektem /np. w płatności/.Atrybuty są podstawą układów klasyfikacyjnych w systemie informacyjnym. Według atrybutów identyfikacyjnych sporządzane są np.rozdzielniki kosztów, kartoteki ewidencyjne stanów i obrotów. Atrybut czasu jest wykorzystywany do rozdzielania zdarzeń dotyczących różnych okresów ewidencyjnych i sprawozdawczych. Właściwości obiektu są szczególnie ważne w systemach wyszukiwania informacji i opisywane są przez tzw. deskryptory.

W systemie informatycznym strumień zdarzeń informatycznych występuje w postaci zbiorów transakcyjnych i podczas przebiegów aktualizacji dociera do charakterystyk obiektów znajdujących się w zbiorach głównych. Zwykle zdarzenia są rozproszone po wielu zbiorach transakcyjnych, aczkolwiek w pewnych warunkach możliwe jest zgrupowanie ich w jednym strumieniu wejściowym. Za warunki te uważać można:

- występowanie procedur wielokrotnego użytku /tzn.procedur służących do obsługi wielu typów zdarzeń/,
- zastosowanie jednolitej symbolizacji zdarzeń, niezależnej od tego w jakim zbiorze transakcyjnym zdarzenie dotarło do systemu /metoda taka została przez autora przedstawiona w pracy /39, s.131-138/.
- wykorzystanie mechanizmów dynamicznego przywoływania i montażu procedur, stosownie do potrzeb obsługi rozpoznanego zdarzenia.

Warunek ostatni ma przeciwdziałać tworzeniu monstrualnie dużych programów, zajmujących nadmiernie pamięć operacyjną i trudnych do modyfikacji. Nie zawsze systemy operacyjne ułatwiają realizację tego warunku, ponadto w tradycyjnych systemach nie sprzyja temu występowanie wielu zbiorów głównych /zamiast kilku baz danych/. Tryb przetwarzania, w którym występują te trzy warunki, nazwać można przetwarzaniem sterowanym przez transakcje. Przy zorientowaniu systemu na transakcje /zdarzenia/ stosowane być mogą różnego rodzaju uniwersalne dokumenty obrotu zasobami materialnymi oraz łatwiej zawiązywane struktury logiczne danych opierające się na relacjach pomiędzy zdarzeniami /zamówienie, dostawa, zapłata, itp./. Co więcej, zdarzenia przestaną być tylko krótkotrwałym wejściem, lecz będą przetwarzane na takich samych prawach jak zapisy obiektów w zbiorach głównych.

Technologia przetwarzania sterowanego transakcjami sprzyja rozwojowi systemu, gdyż w przypadku wystąpienia nowego zdarzenia, opatrzonego symbolem rozpoznawczym według jednolitej klasyfikacji zdarzeń w organizacji /przedsiębiorstwie/, rozpływ danych pierwotnych zawartych w zdarzeniu przebiega według normalnego trybu, np. drogą konwersji symbolu na adres dyskowy rekordu bazy danych. Podstawowy "wsad" programowy obsługi zdarzeń pozostaje bez zmian, natomiast modyfikacji wymagać będzie schemat bazy danych, między innymi w zakresie opisu pól i powiązań logicznych. Ten tryb przetwarzania ułatwi modelowanie rzeczywistości gospodarczej, gdyż może dostarczać modelom dane łatwo lokalizowane w czasie i przestrzeni. Wierniejsze staje się więc odwzorowanie dynamiki zmian niż w przypadku operowania danymi zagregowanymi pochodzącymi ze zbiorów głównych.

Reasumując rozważania na temat inspiracji płynących z teorii zdarzeń do metodologii projektowania strukturalnego, sformułować trzeba wniosek, iż przerzucają one akcent z "dobrze" /na stałe/ ustrukturalnionych systemów informatycznych na rozwiązania rozwijalne z "płynną" technologią przetwarzania danych. Ponadto teoria zdarzeń ułatwia poznawanie problemów, gdyż preferuje zasadę pierwszeństwa czasu nad przestrzenią. Zasada ta oznacza, że własności obiektów powstają w ich rozwoju, a rozwój składa się ze zdarzeń. Obiekty więc, jako przestrzeń, są od zdarzeń /czyli czasu/ zależne.

W tym samym kierunku na projektowanie strukturalne oddziałuje teoria przypadkowości w zarządzaniu /contingency theory of management/ sformułowana przez Lawrence'a i Lorscha /32/, którzy badając struktury organizacyjne stwierdzili, że powinny one wynikać ze specyficznych /konkretnych/ potrzeb środowiska zewnętrznego i nie mogą być tak zaprojektowane aby spełniać wymagania wszystkich sytuacji. W myśl tych twierdzeń system nie powinien być strukturyzowany w pełni na podstawie np. długoterminowego planu rozwojowego, lecz dynamicznie w sposób odpowiadający problemom, które przypadkowo /nieoczekiwanie/ pojawiają się, mimo iż nie ujęto ich w scenariuszu działania. Wychodząc z tej teorii, tzw. systemowe analizy przedsiębiorstw stosowane w projektowaniu klasycznych systemów informatycznych są mało przydatne, gdyż cele i komponenty określone w momencie analizy nie odpowiadają sytuacjom przy wdrażaniu systemu.

Z teorią przypadkowości ściśle związana jest teoria sytuacyjnego zarządzania /situational theory of management/. Jej przedstawiciel, R.J. Mockler, /33/ twierdzi, że każda konkretna sytuacja powinna być indywidualnie rozpatrywana pod kątem doboru do niej potrzebnej teorii i metody zarządzania.

Mimo, iż obie te teorie powstały na gruncie teorii zarządzania i organizacji, formułują one również kluczową zasadę projektowania strukturalnego systemów informatycznych, głoszącą iż podstawy struktury systemu działania nie stanowią uprzednio spreparowane teorie i struktury, lecz problemy i sytuacje.

Same zasoby instrumentalne /teorie, metody/ nie stanowią jednak wystarczającego elementu sprawności systemu. Niezbędnym ogniwem tego systemu jest człowiek i grupy społeczne, w których działa. Zwracają na to uwagę takie podejścia projektowe jak socjo-techniczne /socio-technical approach/ i współangażujące /participative approach/ rozwijane ostatnio w krajach Europy Zachodniej.

W podejściach tych szczególną uwagę zwraca się na wielowymiarowe traktowanie środowiska, w którym znajduje zastosowanie system informatyczny:

- analizuje się nie tylko formalną strukturę organizacyjną, lecz również nieformalne stosunki istniejące w organizacji;

często twierdzi się nawet, że te ostatnie są ważniejsze, aby poznać istotę procesów zachodzących w niej,

- uwzględnia się nie tylko działania racjonalne, lecz też irracjonalne,
- rozpatruje się czynnik ludzki zarówno w skali jednostki jak i grup społecznych /stowarzyszonych w związkach zawodowych, organizacjach politycznych, itp./,
- bierze się pod uwagę zarówno czynniki techniczno-ekonomiczne jak i socjalne /np. wprowadzenie komputerowego systemu przetwarzania danych traktowane jest często jako zmiana warunków pracy/.

Podejście takie stawia system informatyczny w specyficznej sytuacji, po pierwsze, traktując go jako jedno ze źródeł informacji /tej formalnej/, zaś po drugie, narzucając mu wymagania zarówno użytkowników indywidualnych jak i grup społecznych /sformułowane niekiedy w ustawodawstwie państwowym/.

Czołowy reprezentant podejścia współangażującego E. Mumford /35/ widzi w użytkowniku stronę odpowiedzialną za system oraz współrealizatora wszystkich etapów budowy systemu. Uważa, iż jedynie w ten sposób można zbudować system odpowiadający potrzebom użytkowników oraz uzyskać ich poparcie jako współautorów.

Przedstawiciel podejścia socjo-technicznego A. de Maio /34/ bierze w obronę system organizacji uważając, że nie powinien być on naginany do sztywnej technologii partiowo-okresowego przetwarzania.

Oba podejścia preferują więc punkt widzenia użytkownika, chroniąc jego potrzeby informacyjne i swobodę organizacyjną. Realizowane mogą być jednak jedynie wtedy, gdy odpowiada im równie elastyczna technologia przetwarzania danych. Metoda strukturalnego projektowania zmierza właśnie w kierunku spełnienia tych wymagań. Być może doprowadzi ona w końcowym etapie swego rozwoju do takich narzędzi projektowo-programowych, że każdy użytkownik będzie mógł formułować swoje potrzeby w dostatecznie prostym języku specyfikacyjnym, następnie otrzymać zestaw wygenerowanych automatycznie procedur i sprawdzić je na własnych danych testowych lub wygenerowanych przez system.

W rozpoznaniu potrzeb użytkowników strukturalne projektowanie otrzymuje pomoc metodologiczną od infologicznej metody projektowania, opisanej przez P.Kerolę w /31/. Główny wysiłek w tej metodzie skierowany jest na identyfikowanie informacji, ocenę jej znaczenia /wpływu/ oraz logiczne - niezależne od komputerowego procesu - definiowanie jej zawartości. Obiektem zainteresowania w podejściu infologicznym stają się ludzie i organizacje użytkujący informacje, kryteria jej użytkowania oraz stopień użyteczności systemu i jego oddziaływanie na środowisko które obsługuje.

Pisząc o genezie strukturalnego projektowania, nie można ograniczyć się do ogólnych i szczególnych metod projektowania, pomijających wykonawcze aspekty programistyczne. Nie sposób bowiem zapomnieć, że powstaje ono w dużej mierze pod bezpośrednim wpływem modularnego /modułowego/ i strukturalnego programowania. Zasady i reguły tych metodycznie rozpracowanych technik programistycznych będą przewijać się w niniejszej pracy, jednakże skala problemów jakie są przez nie realizowane nie pozwala na traktowanie ich jako wystarczających narzędzi konceptualnych projektowania systemu. Wobec poważnych rozbieżności w kwestii definicji modularnego i strukturalnego programowania, nie będziemy w stanie - w ramach tego opracowania - dostarczyć Czytelnikowi jednoznacznej i pełnej ich charakterystyki. Przytoczymy jedynie te stanowiska, które wydały się nam interesujące z punktu widzenia przedmiotu pracy.[1/]

W dokumencie Diebolda E53a /11, s.46-51/ programowanie modularne uważane jest za strukturalną /nie algorytmiczną/ teorię programowania, jako metoda podziału programu na takie jednostki logiczne zwane modułami, że każdy z nich może być napisany/kodowany/, testowany, korygowany, modyfikowany /itd./ niezależnie od innych. Wśród modułów wyróżnia się moduł sterujący, moduły przetwarzania, moduły wielokrotnego użytku i moduły wejścia-wyjścia. Moduł sterujący nie wykonuje sam żadnych zadań przetwarzania,

---

1/ Czytelników szerzej zainteresowanych tą tematyką odsyłamy do literatury, której wykaz zamieszczamy na końcu opracowania /6, 55, 56, 65/.

zawiera między innymi obszary i dane wspólne dla modułów przetwarzania oraz rozdziela zadania dla modułów przetwarzających. Każdy moduł przetwarzania jest wyodrębnioną jednostką, nie wpływającą na przetwarzanie w innym module, wykonującą zadania przetwarzania, dla którego zawiera dane i obszary robocze. Moduły wielokrotnego użytku są to sekwencje instrukcji używane w kilku modułach przetwarzania, zaś moduły wejścia-wyjścia realizują specjalne funkcje dotyczące zbiorów /np. przetwarzanie etykiet/. Moduły powinny komunikować się poprzez określone bramy komunikacyjne, wśród których na podkreślenie zasługuje mechanizm formalnego parametru procedury.

E. Yourdon twierdzi, że programowanie strukturalne jest pogłębieniem programowania modularnego i polega na sprowadzeniu programu - poprzez stopniową dekompozycję z góry do dołu /top-down/ - do niepodzielnych modułów "atomowych", którymi są poszczególne zdania obliczeniowe, zdania selekcji i pętle. W każdym module obowiązuje zasada jednego wejścia i jednego wyjścia, zdania selekcji IF-THEN-ELSE mogą być dowolnie zagnieżdżane, zaś pętle powinny być realizowane za pomocą wyrażenia DO-WHILE /ALGOL, PL/1/ lub PERFORM-UNTIL /COBOL/ /65, s.148-149/.

Podstawową zasadą programowania strukturalnego jest unikanie skoków GO TO /12, s.52/ lub nawet ich całkowite wyeliminowanie /8/. W liberalnej konwencji programowania strukturalnego wyraźnie zabroniony jest skok GO TO z jednego składnika programu do innego, zaś dozwolony tylko w ramach jednego modułu lub do wyprowadzenia sterowania na przetwarzanie sytuacji wyjątkowych. Wyeliminowanie GO TO w maksymalnym stopniu ogranicza przeplatanie się dróg logicznych w programie /nie występują nawroty z dołu do góry/, co posiada niebagatelne znaczenie w kontekście uruchamiania i modyfikowania programów.

Za ojca strukturalnego programowania uważany jest Edsger W. Dijkstra, profesor Uniwersytetu w Eindhoven, który już w roku 1965 na Kongresie IFIP w Nowym Jorku /7/ postulował wyeliminowanie zwrotu GO TO, a następnie powtórzył swoją propozycję w 1968 roku w liście do redaktora Comm. of ACM /8/, zaś w cztery lata później wspólnie z Dahlem O.J. i Hoarem C.A.R. opracował monografię na temat strukturalnego programowania /6/. Prekursorami

mechanizmów strukturalnego programowania byli C. Böhm i G. Jacobini, którzy w 1966 roku udowodnili /2/, że program może być zbudowany z dwóch /poza sekwencją/ mechanizmów sterujących:pętli DO-WHILE i binarnych decyzji /selekcji/ IF-THEN-ELSE.

Programowanie modularne i strukturalne stanowi istotny punkt wyjściowy do strukturalnego projektowania elementów programowych,stanowiących w przypadku systemów przetwarzania danych bardzo liczebną i równocześnie trudną do skoordynowania /rozproszoną/ część składową systemu. Na gruncie technik modułowego i strukturalnego programowania, ograniczonych w zasadzie do aparatu formalnego języków programowania, nie można rozwiązać kwestii strukturyzacji systemu na poziomie projektowania, która wymaga nie tyle mechanizmów sterujących fragmentami systemu, ile mechanizmów ich łączenia. Ponadto dochodzi do tej oceny również czynnik organizacyjny: to, co jest odpowiednie do konstruowania elementarnych bloków wykonawczych przez pojedynczego programistę, nie musi być równie dobre i funkcjonalne dla zespołów wieloosobowych działających na poziomie wyższych jednostek strukturalnych. Stąd powstała potrzeba opracowania metody strukturalnego projektowania, która zmierza do uzyskania odpowiedzi na następujące pytania:

- jakie powinny być kryteria dekompozycji systemu i problemu,
- jaka ma być konstrukcja łącz międzymodułowych,
- jak w ramach łączy rozwiązać problem przekazywania danych i parametrów sterujących,
- jakie zabezpieczyć możliwości przekształcania źródłowych tekstów procedur, aby można było je dostosować do różnych zadań przetwarzania,
- jakie narzędzia dostarczyć projektantom w ramach metody strukturalnego projektowania.