

Ten okres proponujemy przyjąć dla realizacji RSI<sup>121</sup>.

Zgodnie z prawami natury, trwałość „raz zaprojektowanego RSI” nie jest wieczna. Przewiduje się, że po 18—20 latach od podjęcia prac nad systemem — osiągnie on 90% nasycenia użytkowego. W tym czasie, wskutek postępu społecznego, gospodarczego, naukowego itp. trzeba będzie zaprojektować jego następną „generację”. Na rysunku 4.67. podano krzywą życia RSI. Natomiast na rysunku 4.68. podano zachodzenie kolejnych generacji RSI.

Zakłada się, że dotychczasowe prace, tj. przed zastosowaniem omówionej koncepcji można nazwać RSI<sup>0</sup>, będą trwały parę lat, a użytkowanie zaniknie, np. w 5 lat potem, kiedy zaczną się upowszechniać RSI<sup>1</sup> (omówione dotychczas). Stąd wynika dwutorowy tryb realizacji RSI.

## 4.5.

### Problemy projektowania elementów gospodarczych systemów informatycznych

#### 4.5.1.

#### Ewolucja metod projektowania modeli systemów informatycznych dla potrzeb kierowania systemami gospodarczymi

W połowie lat pięćdziesiątych, kiedy zaczęła się krystalizować koncepcja wykorzystywania komputerów w „przetwarzaniu danych” rozpoczęły się próby tworzenia modeli informatycznych dla potrzeb kierowania organizacjami gospodarczymi. Starano się zamodelować podstawowe procesy w ujęciu całościowym i wyrazić je w języku ekonometrycznym. Z uzyskanych wyników wynikała mała przydatność tych modeli w praktyce<sup>122</sup> i co się z tym wiąże, można zaobserwować pewien sceptycyzm u użytkowników owych modeli. Z drugiej strony, trudno zgodzić się z pochodzącą stąd argumentacją, że wobec złożoności procesu kierowania wszelkie próby zastosowań modeli tego typu z góry skazane są na niepowodzenie. W tej sytuacji konieczne staje się określenie kryterium oceny sukcesu zastosowanego wymienionych typów modelu.

<sup>121</sup> Dla ewentualnych oponentów tej propozycji można przypomnieć, że znany z literatury i Programu DIEBOLDA — zintegrowany system informatyczny w Micro-Switch w firmie Honeywell był realizowany ponad 15 lat. Okres 8,5 lat jest krótszy od okresu dwóch pięćdziesiątek, stąd ma charakter bardziej „mobilizacyjny” i kontrolowany.

<sup>122</sup> Por. R. Hayes, R. Nolan, *What Kind of Corporate Modeling Functions Are Best*, „Harvard Business Review” 1974, May-June.

Prosty proces będzie wymagał prostego modelu, ale wcale to nie oznacza, że złożony proces wymaga złożonego modelu. Złożoność procesu wynika głównie z licznych zmian relacji między składnikami procesu. Stąd model informatyczny dla takiego procesu musi być nie tyle złożony, co dostosowujący się do zmian. Powodzenie zastosowaniowe modelu można także mierzyć okresem jego wykorzystywania.

Ewolucję zastosowań modeli informatycznych można scharakteryzować według stosowanych metod projektowania. Wyróżnimy następujące metody projektowania (według kryterium zakresu rozpatrywanych obiektów, procesów, zasobów i funkcji informacyjnych i informatycznych):

- 1) projektowanie wycinkowe (*bottom up* — oddolne) od 1956 r.,
- 2) projektowanie kompleksowe (*top down* — odgórne) od 1961 r.,
- 3) projektowanie całościowe, od 1971 r.

*Projektowanie wycinkowe* było źródłem sukcesu informatyki, bowiem użytkownik mógł łatwo porównać wydruki komputerowe dotyczące płac i księgowości z dotychczasowymi formami wykazów powstałych ręcznie lub przy wykorzystaniu małej i średniej mechanizacji. Do grona użytkowników-księgowych dołączyli technicy i planiści produkcji, którzy zgłaszali potrzeby na modelowanie przebiegów produkcyjnych i dystrybucyjnych. Projektantami modeli byli specjaliści z otoczenia ośrodka obliczeniowego. Na ten obszar zainteresowań modelowych miały wpływ następujące czynniki:

- metody badań operacyjnych były już dostatecznie rozwinięte i stały się podstawą projektowania,

- komórki produkcyjne były dość dobrze rozpoznane pod względem mechanizmów przepływu materiałów i informacji, co gwarantowało względnie łatwe kwantyfikowanie zjawisk w nich zachodzących,

- zdarzenia, które zachodziły w komórkach produkcyjnych, zostały na tyle dobrze rozpoznane, że można było względnie łatwo zaproponować konsekwencje decyzyjne dla poszczególnych zdarzeń,

- częstotliwość występowania zdarzeń w komórkach produkcyjnych była tak wysoka, że użytkownik z zadowoleniem przyjmuje system informatyczny jako pomoc,

- dotychczasowy rozwój metod wytwórczych wytworzył określoną wiedzę techniczną u użytkowników, których (zwykle) wykształcenie techniczne ułatwiało kontakt z informatykami.

Dzięki demonstrowaniu informatycznych modeli produkcyjnych (dotyczących produkcji podstawowej) kadrze kierowniczej wyższych szczebli powstawały nowe wnioski dotyczące rozszerzania zakresu modelu o pozostałe procesy produkcyjne i funkcje informacyjne oraz ujęcie w nim wszystkich jednostek organizacyjnych. W konsekwencji uważano, że tak powstały model będzie wymagał większych komputerów. Pewne próby w tym

zakresie powstały w przemysłach lekkich (w których występuje możliwość substytucji środków pracy i materiałów)<sup>123</sup>. Z chwilą podejmowania prób wykorzystania tych modeli do planowania wieloletniego okazało się, że trzeba było przepracowywać modele, trudno adaptujące się przy zmieniających się celach i warunkach.

Ponieważ narzędzia informatyczne (języki) były w owym czasie jeszcze na niskim poziomie, a dane trzeba było *ad hoc* zbierać (dziś można by je pobierać ewentualnie z banków informacji) — zatem niepowodzenie owych modeli było oczywiste. Projektant wówczas stosował podobny rodzaj gry jak użytkownik: wyglądał model, raz dodając mu „rzeczywistości”, a raz poszukując w nim „precyzji”.

W wyniku tej gry wydłużał się okres, w którym model mógł dostarczyć wyniki użytkownikowi. Zleceniodawcy wycofywali swe poparcie i w ten sposób stopniowo rezygnowano z posługiwania się modelami.

Na podstawie tej oceny można sformułować następujące spostrzeżenia:

— modele operacyjne (regulujące przebieg operacji) nie są modelami sterowniczymi; różnią się rodzajem informacji WE i WY oraz procedurami,

— modele sterownicze wymagają od projektantów specjalizacji i wiedzy określonej dla danego obszaru funkcji kierowania.

Rozwój metod projektowania kompleksowego wynika m.in. z dwóch czynników. Rozwój narzędzi informatycznych do formy sprzętu tzw. III generacji (pojemne pamięci wyrywkowe, zdalne przetwarzanie, niezawodność itp.) stworzył po 1969 r. techniczne warunki do projektowania i stosowania rozwiniętych modeli systemów informatycznych. Zmienił się również skład realizatorów owych systemów. Środowisko użytkowników wykształciło lub wyszkoliło w tym zakresie swoich członków, których charakteryzuje wspólny język oraz znajomość modelowanych zjawisk. Zauważono ograniczone możliwości zastosowania modeli operacyjnych do procesów sterowania ekonomicznego i planowania. Rozpoczęły się poszukiwania koncepcji wymienionych modeli w warunkach zdecydowanego oporu ze strony średniego szczebla pracowników. Został, niesłusznie, wytworzony klimat zagrożenia dla nich. W wyniku projektowania kompleksowego powstawały rozbudowane modele systemów, produkujące zwały wydruków, które nie cieszyły się dużą poczytnością. Ponadto wystąpiły ograniczenia w zbieraniu danych na ogół nieaktualnych i niejednorodnych.

Wskutek rozległości modeli i ich problematycznej wdrażalności projektanci tracą zleceniodawców. Okazuje się, że w pewnym momencie ni-

<sup>123</sup> W Polsce prace w tym zakresie prowadzili m.in. M. Lesz i W. Radzikowski w połowie lat sześćdziesiątych.

komu nie zależy na uruchomieniu modelu S/I, i że najbardziej zainteresowanym w tym jest sam autor.

Wylaniają się stąd następujące spostrzeżenia:

1) samo poparcie najwyższego kierownictwa nie jest wystarczające do osiągnięcia sukcesu wdrożeniowego; kierownictwo musi rozumieć istotę modelu, być nawet jego współautorem,

2) wielkie, wszystko zawierające modele stwarzają problemy typu projektowego i informacyjnego, trudne są w zrozumieniu, rozumieją je (tylko czasem w pełni) sami autorzy.

Projektowanie całościowe rozwinęło się z chwilą podjęcia prac nad wielkimi systemami informacyjnymi. W Polsce po raz pierwszy w 1971 r., kiedy została wysunięta koncepcja i projekt ogólny Krajowego Systemu Informatycznego<sup>124</sup>. Z tej koncepcji wyniknęły rozwiązania pilotowe krajowych systemów informatycznych typu: WEKTOR, PESEL, MAGISTER, CENPLAN, SPIS, TRAKT, RSI (MPM) i INFOSTRADA. Podobne prace podjęto także w ZSRR nad systemem OGAS i pochodnymi<sup>125</sup>. W innych krajach metoda ta nie rozwinęła się ze względu na niedojrzałość sytuacji lub na ograniczenie wynikające z koncepcji społeczno-politycznych.

Metoda projektowania całościowego polega na uwzględnieniu wszystkich typowych obiektów, dla których projektuje się S/I, z tym zastrzeżeniem, że zwraca się przede wszystkim uwagę na klasyfikację podsystemów i sposoby ich współdziałania. Szczegółowe rozwiązania podsystemów są wyznaczone wymaganiami zewnętrznymi totalnego S/I oraz wewnętrznymi danego podsystemu. Metodę tę można porównać z podejściem urbanistycznym, w którym na plan pierwszy wysuwa się sprawę koncepcji przestrzennego zagospodarowania i lokalizacji obiektów realizujących określone funkcje urbanistyczne — podczas gdy rozwiązania szczegółowe projektowane są z chwilą realizowania poszczególnych inwestycji.

W metodzie projektowania całościowego wykorzystuje się tzw. podejście systemowe. Chodzi w nim o uchwycenie podstawowych mechanizmów wewnętrznych danego systemu, które umożliwiają rozstrzygnięcie prowadzące do uproszczenia modeli (bez tracenia ich zalet). Stąd duży nacisk kładzie się na zagadnienia logiki systemu.

Z logiki systemu winny wynikać propozycje opisu metodami formalnymi, a nie odwrotnie. Stąd staje się konieczne sięganie do elementarnych prawideł i definicji organizacji i kierowania, których dotychczasowe sformułowania były jeszcze przydatne w metodach projektowania wycinkowego i kompleksowego.

<sup>124</sup> Por. A. Targowski, *Organizacja procesu przetwarzania danych*, wyd. cyt.

<sup>125</sup> Przegląd dorobku światowego w tym zakresie, por. A. Targowski, *Organizacja procesu przetwarzania danych*, wyd. cyt.

Tablica 4.7.

Podział cyklu rozwojowego systemu informatycznego na fazy i etapy według wybranych autorów

	PRZYGOTOWANIE	URUCHOMIENIE I	EKSPLOATACJA	URUCHOMIENIE DOSKONA- LACE	EKSPLO- ATACJA PO I UDOS- KONALE- NIU
Według C. Burilla, L. Ellswortha	1. Szkolenie 2. Planowanie 3. Projektowanie ogólne: — identyfikacja problemu — specyfikacja wymagań — określenie rozwiązania — akceptacja użytkownika	4. Projektowanie szczegółowe 5. Programowanie 6. Testowanie 7. Instalowanie 8. Ocena prawidłowości uruchomienia 9. Akceptacja użytkownika	— kierowania systemem — analiza eksploatacyjna — wnioski usprawniające — akceptacja przez użytkownika	projekty usprawniające itd.	Kierowania systemem
Według Z. Gackowskiego	1. Stadium wstępne: — potrzeby, — zadanie projektowe 2. Projektowanie wstępne: — wybór rozwiązań, — założenia techniczno-ekonomiczne 3. Projektowanie techniczne: — szczegółowe projektowanie, — programowanie — testowanie. 4. Przygotowanie dokumentacji inwestycyjnej	Realizacja: 5. Rozruch 6. Wdrażanie	7. Eksploatacja		
Według A. Targowskiego	1. Analiza systemu 2. Założenia systemu 3. Projekt wstępny 4. Projekt techniczny (z programami)	5. Projekt techniczno-roboczy — dokumentacja kontroli, — dokumentacja eksploatacji, — dokumentacja instrykcyjna	6. Eksploatacja 7. Zmiany	8. Konserwacja 9. Zmiany	

Źródło: opracowane na podstawie prac C. Burilla, L. Ellswortha, IBM System Science Institute, New York; Z. Gackowskiego, *Projektowanie systemów informacyjnych zarządzania*, wyd. cyt.; A. Targowskiego, *Automatyzacja przetwarzania danych*, wyd. cyt.

## 4.5.2.

## Ewolucja technik projektowania modeli systemów informatycznych dla potrzeb kierowania systemami gospodarczymi

Badanie systemu informatycznego według faz rozwoju, tj. według modelu  $M^{VII}$  prowadzi do koncepcji cyklu rozwojowego S/I, w którym wyróżnia się następujące fazy: przygotowanie, uruchomienie, eksploatację, doskonalenie i wymianę systemu.

W tablicy 4.7. podano podział cyklu rozwojowego S/I na fazy i etapy według niektórych autorów. Na rysunku 4.69. podano systemowe podejście do projektowania<sup>126</sup>. Na tle różnych podejść do sprawy projektowania S/I można wyróżnić etapy o charakterze proceduralnym i projektowym. Na przykład do pierwszego rodzaju etapów można zaliczyć: szkolenie, planowanie, odbiory, instalowanie, oceny itp. Do drugiego rodzaju etapów można zaliczyć: analizę dotychczasowego systemu (I), ocenę wymagań nowego systemu (II), projektowanie systemu (III), programowanie i testowanie (IV), eksploatację (V), konserwację i zmiany (VI). Według tak wyróżnionych etapów cyklu rozwojowego S/I zbadamy ewolucję technik projektowania S/I.

Na rysunku 4.70. podano schemat rozwoju technik analizowania systemów informacyjnych<sup>127</sup>. Wśród nich można wyróżnić trzy kierunki:

- 1) analizowanie przepływu informacji technikami graficznymi,
- 2) analizowanie problemów decyzyjnych technikami badań operacyjnych i statystycznymi,
- 3) analizowanie wykorzystania zasobów.

Główną uwagę ześrodkujemy na kierunku pierwszym. Do 1910 r. dorobek w tym zakresie został wypracowany przez F. Taylora, H. Ghantta, F. Gilberta, K. Adamieckiego i innych, którzy wprowadzili różnego rodzaju techniki harmonogramowania.

W okresie 1920—1950 (kierunek drugi) nastąpił rozwój małej, średniej i dużej mechanizacji, który spowodował pojawienie się technik graficznego opisu relacji dokumentów i urządzeń mechanizujących ich obróbkę. Jeszcze na początku rozwoju zastosowań komputerów, ten rodzaj technik był stosowany. Schematy tego typu charakteryzowała ociążałość i rozdrobnienie aż do najmniejszych szczegółów, co nie pozostało bez wpływu na ograniczanie ogólnego spojrzenia na cały system.

W okresie 1950—1960 (kierunek trzeci) powstały techniki przystosowane do narzędzia, jakim stał się komputer. Powstały tzw. schematy

<sup>126</sup> Rysunek pochodzi z pracy B. Pełki, *Zarys ekonomiki i organizacji przemysłowych procesów produkcyjnych*, wyd. cyt.

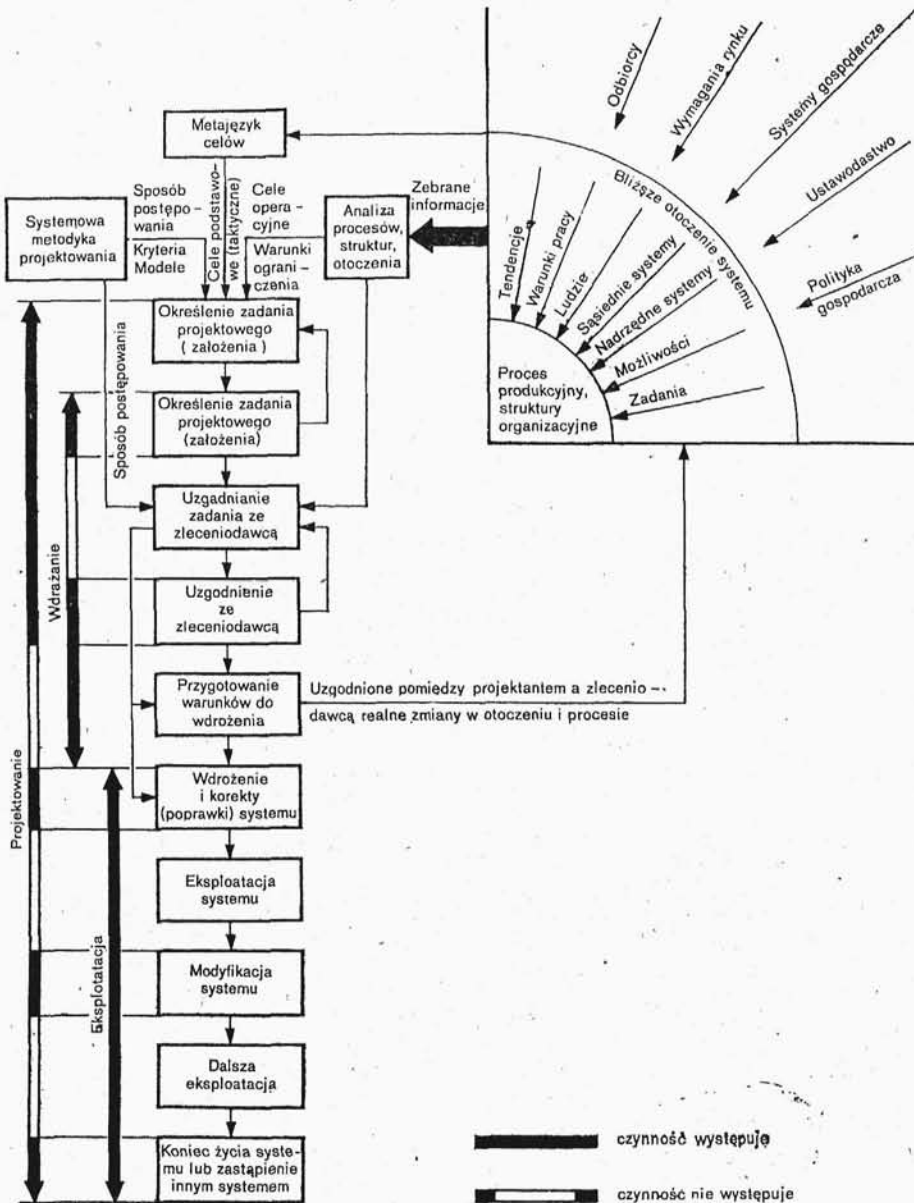
<sup>127</sup> Rysunek pochodzi z pracy I. D. Congera. Por. I. D. Conger, *Evolution of Business System Analysis Techniques*, „Computing Surveys” 1973, no 3, vol. 5, September.



blokowe, których uogólnienia stanowiły organigramy (tzw. flowchart). Z połączenia obu technik powstała technika „kart obiegu informacji” obrazująca WE, WY, użytkowników i operacje wykonywane przez nich na

**Rysunek 4.69.**

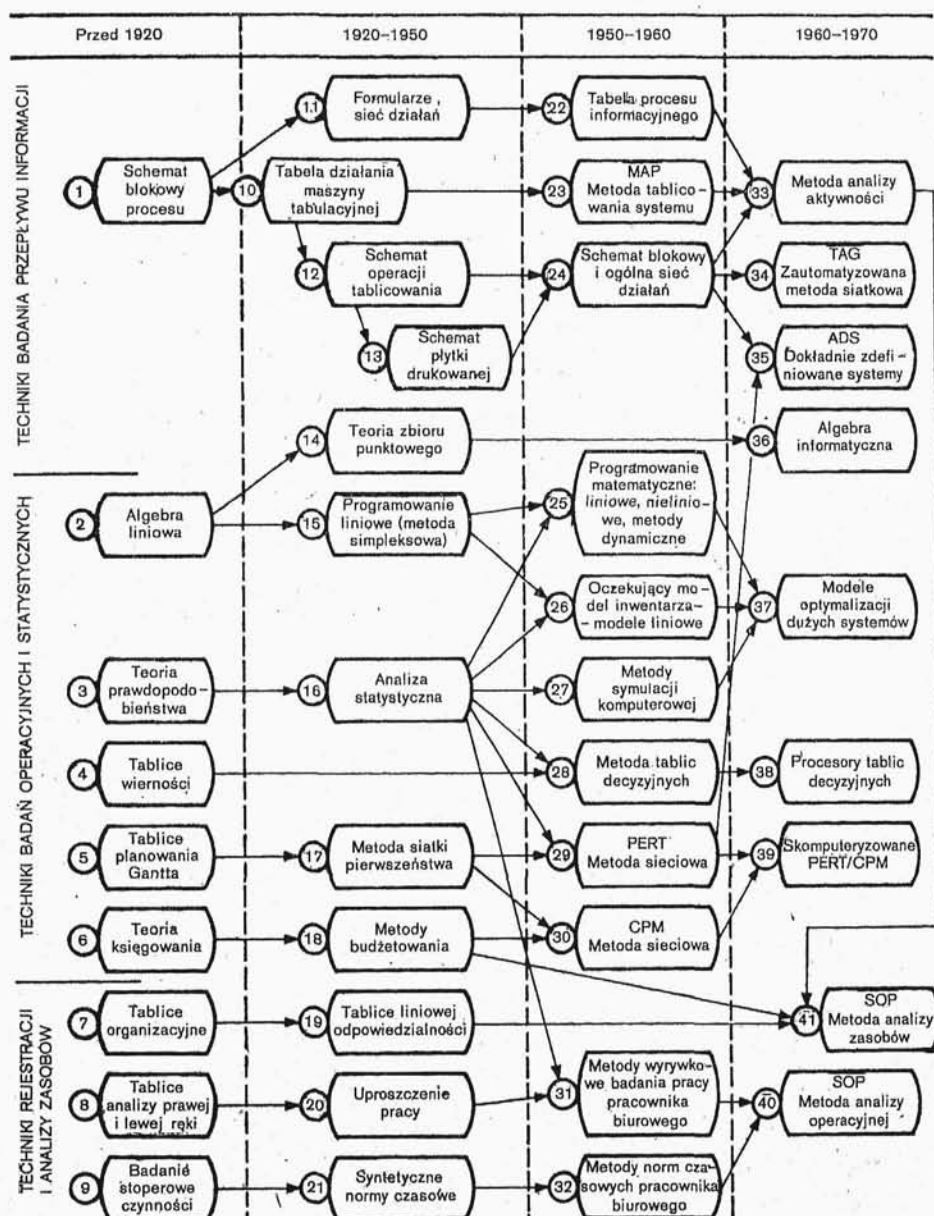
Systemowe podejście do projektowania podsystemu „planowanie produkcji i kontrola międzywydziałowa”



informacjach. W toku licznych prac tego typu wypracowano konwencje dotyczące słów i symboli, które pozwalały na porównywanie schematów. Można tu wymienić technikę MAP wypracowaną przez firmę NCR.

Rysunek 4.70.

Rozwój ważniejszych technik analizy systemowej





W latach 1960—1970 kilka firm produkujących komputery zadbało o techniki projektowania S/I. Wśród bardziej popularnych można wymienić:

1. ADC (*Accurately Defined System*) firmy NCR polega na posługiwaniu się pięcioma wzorami formularzy dla opisu: WY, WE, procedur przetwarzania, procedury pozainformatyczne, sytuacji decyzyjnych (w tablicach decyzyjnych). W ramach tego systemu zostały zapewnione powiązania między poszczególnymi informacjami. Dzięki temu omyłki, opuszczenia itp. błędy można łatwo wykryć.

2. Algebrę informacyjną opracowaną przez R. Bosaka w ramach grupy CODASYL<sup>128</sup>. Chodziło o stworzenie techniki niezależnej od sprzętu, a będącej na poziomie S/I. Wykorzystując teorię zbiorów została zaproponowana technika określania relacji procedur na danych. Projektant miałby określać własności zbiorów danych i reguły powstawania nowych. Pozostałe funkcje opisu przetwarzania miały być pozostawione programowi kompilującemu. Niestety grupa CODASYL nie wypracowała języka użytkownika, ani wspomnianego kompajlera. Jednakże koncepcje algebry informacyjnej zostały później wykorzystane (o czym będziemy jeszcze mówić).

3. SOP (*Study Organization Plan*) firmy IBM polega na łączeniu kilku technik w jedną. Składa się z analizy: wykorzystania zasobów, czynności (procesów) i przepływu informacji (komunikatów i zbiorów) na odpowiednio zbudowanych formularzach.

4. BSP (*Business System Planning*) firmy IBM polega na rozwinięciu techniki SOP w kierunku:

- a) zwrócenia większej uwagi na analizę działalności gospodarczej,
- b) wyłaniania tych elementów systemu informacyjnego, które wspomagają kluczowe obszary działań gospodarczych,
- c) budowy ogólnej sieci informacyjnej,
- d) wyłaniania „pierwszego najbardziej potrzebnego” podsystemu informacyjnego.

Warto dodać, że techniki: ADS, Algebry Informacyjnej dotyczą tylko etapu I i II. Do tej grupy można zaliczyć technikę ARDI (*Analysis, Requirements, Determination, Design and Development, Implementation, and Evaluation*) opracowaną przez firmę Philips. Techniki wymienione są szczególnie przydatne w projektowaniu wycinkowym. Podczas gdy technikę SOP można wykorzystywać w metodzie projektowania kompleksowego, a technikę BSP w metodzie projektowania totalnego.

W wymienionych technikach analizy systemów informacyjnych uwagę koncentruje się na podsystemach ewidencyjnej funkcji informacyj-

<sup>128</sup> Por. Codasyl Development Committee, *An information algebra*, „Communication” 1962, AcM 5,1, April.

nej. Dopiero pewnym odstępstwem tu stały się techniki SOP i BSP. Dalejszy rozwój technik projektowania S/I po 1970 r. poszedł w kierunku zastosowania technik komputerowych do projektowania S/I.

Pierwsze zastosowania miały miejsce w dotychczasowych technikach projektowania S/I. Przede wszystkim została podjęta próba z informatyzowania tablic decyzyjnych. Podjęta została w 1959 r. przez grupę CODASYL, która zaproponowała w 1962 r. technikę DETAB-X. Oparto ją na języku COBOL-61, który w tym celu został uzupełniony specjalnym dodatkiem.

W 1965 r. Grupa Specjalnego Zainteresowania Językami Programowania Amerykańskiego Stowarzyszenia Informatyków (ACM-SIGPLAN) opracowała procesor DETAB/65, który akceptuje opisy tablic decyzyjnych w języku COBOL i następnie tłumaczy je na kod źródłowy. Zostało wykonanych szereg implementacji DETAB/65 na maszyny CDC i IBM.

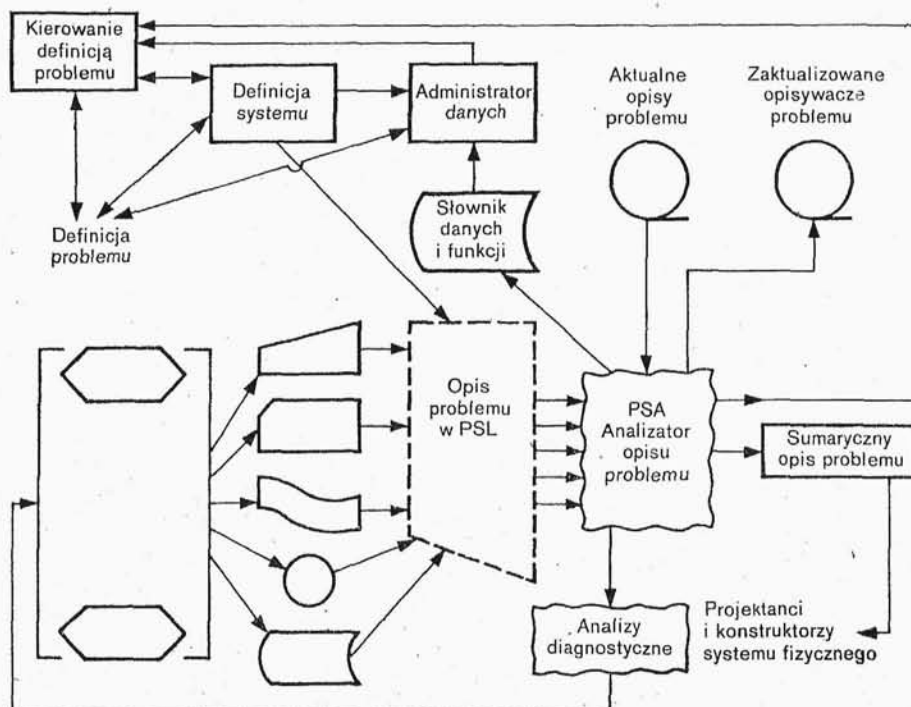
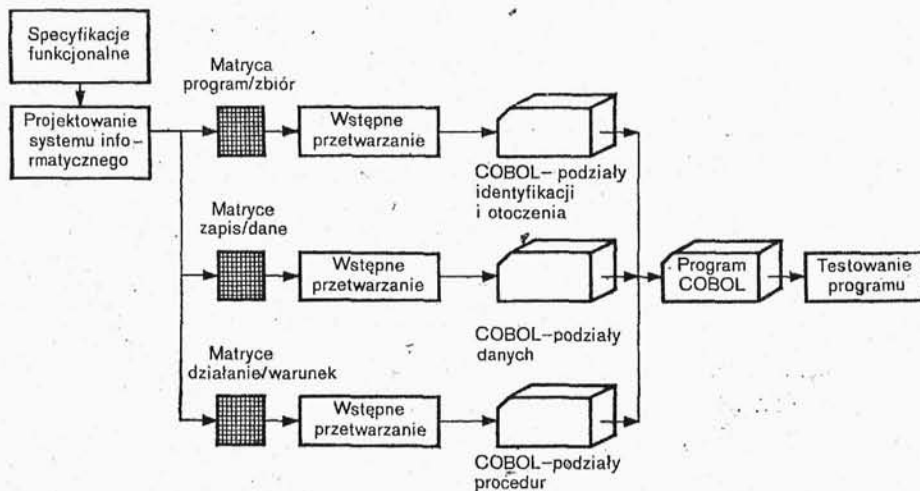
Technika DETAB/65 nie ułatwia pracy projektantom, wpływa natomiast na skrócenie całego cyklu projektowania, szczególnie na etapie wprowadzania zmian.

Technika ADS opracowana pod koniec lat sześćdziesiątych rozwija technikę tablic decyzyjnych w zakresie badania związków informacyjnych występujących przy przejściu od WY do WE. Ponadto opracowuje wykaz i znaczenie nazw oraz diagnostykę.

W 1962 r. powstała w IBM System Research Institute — Technika TAG (*Time — Automated Giret*), którą skomputeryzowano w 1966 r. Podobnie jak ADS wychodzi od analizy WY i dopasowuje odpowiednie WE w odpowiednim czasie, eliminuje przy tym redundancję. Kiedy są zdefiniowane WE i WY, wówczas przystępuje do określenia minimalnej zawartości zbiorów kartotekowych oraz harmonogramów montowania agregatów informacyjnych. Szereg tabulogramów diagnostycznych ułatwia projektantowi analizowanie poprawności projektu torów, kanałów i sieci informacyjnych.

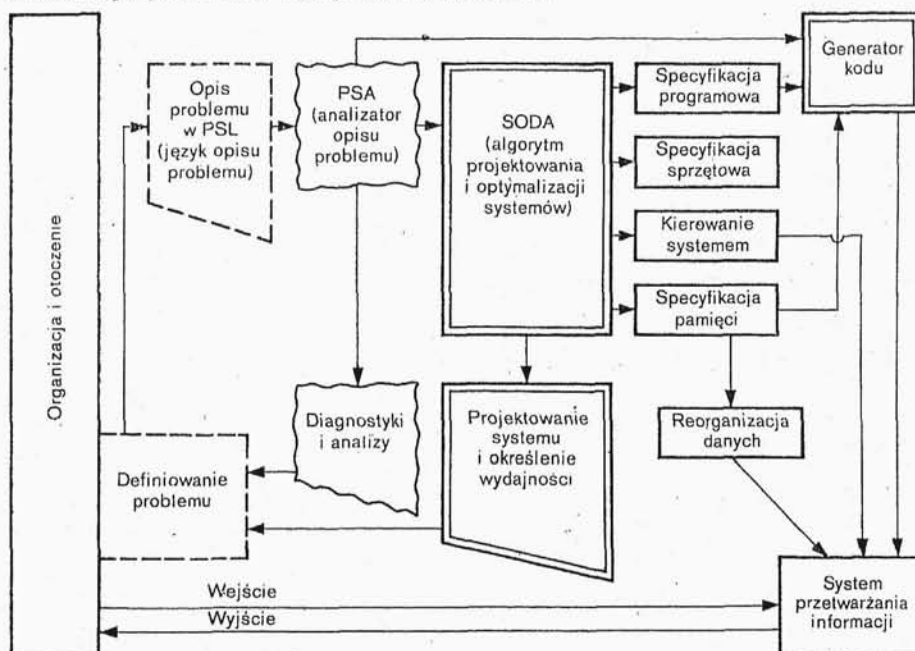
Na bazie algebry informacyjnej, ADS i TAG powstała technika PSL/PSA (*Problem Statement Language*) *Problem Statement Analyzer*, opracowana na Uniwersytecie Michigan. W języku PSL opisuje się WY, parametry WE oraz warunki powstawania WY, podczas gdy PSA po wczytaniu opisu w PSL analizuje wewnętrzne związki informacyjne i wydrukowuje charakterystykę i diagnostykę sieci informacyjnej. Technika jest szczególnie przydatna do aktualizacji dokumentacji opisu systemu. Na rysunku 4.71. podano schemat ideowy PSL/PSA.

Połączenie trzech etapów projektowych (wymagania, projekt i program) w jeden zostało zrealizowane w angielskiej technice Hoskynsa. Ze specyfikacji systemu powstają programy. Na rysunku 4.72. podano schemat działania techniki Hoskynsa.

**Rysunek 4.71.***Język opisu i analizator problemu informacyjnego (PSL/PSA)***Rysunek 4.72.***Technika projektowania i programowania Hoskynsa*

Optymalizowanie wewnętrznych rozwiązań systemu informacyjnego zapewnia technika ISDOS (*Information System Design and Optimization System*), którą łączy się z techniką PSL/PSA, techniką SODA (*System Optimization and Design Algorithm*). Technika SODA wytwarza się programy, które zostają dopasowane do aktualnych warunków pamięciowych. Dzięki temu powstaje program wynikowy (*object*) zdolny do konwersji WE na WY. Na rysunku 4.73. podano schemat działania ISDOS.

**Rysunek 4.73.**  
*Technika projektowania i programowania ISDOS*



Wykorzystanie różnych technik projektowania S/I podano na rysunku 4.74., z którego wynika, że dotąd aktualne w informatyce powiedzenie, że „dzieci szewca chodzą bez butów” — powoli zaczyna nie odpowiadać stanowi faktycznemu.

## 4.5.3.

Zarys techniki projektowania stykowego  
„PRO-RES” modeli systemów informatycznych  
dla potrzeb kierowania systemami gospodarczymi

Z dotychczasowego dorobku praktycznego i teoretycznego metod i technik projektowania S/I wynikają następujące spostrzeżenia: