

PIĄTA WIOSENNA SZKOŁA PTI

# METODOLOGIE TWORZENIA SYSTEMÓW INFORMATYCZNYCH A CASE

Organizowana przez POLSKIE TOWARZYSTWO INFORMATYCZNE

Świnoujście, 18-22 maja 1992 r.



PIĄTA WIOSENNA SZKOŁA PTI

# **METODOLOGIE TWORZENIA SYSTEMÓW INFORMATYCZNYCH A CASE**

Organizowana przez POLSKIE TOWARZYSTWO INFORMATYCZNE

Świnoujście, 18-22 maja 1992 r.



## SPIS TREŚCI

Zdzisław Szyjewski

Od metodologii tworzenia systemów informatycznych do CASE  
(zamiast słowa wstępnego)

Wacław Iszkowski (Digital Equipment Polska)  
Informatyzacja sukcesem biznesu

Piotr Sobolewski (Digital Equipment Polska)  
Środowisko COHESION

Zbigniew Antoni Piotrowski (CasePol)  
ICASE - Zintegrowane środowisko CASE

Roman Lowkis (Computer Systems for Business International Ltd)  
Relacyjna baza danych PROGRESS

Stanisław Wrycza (Uniwersytet Gdański)  
Metody analizy i projektowania systemów w praktyce

B. Fornalik, D. Puternicki, W. Staniszkis (Zeto-Rodan)  
Inżynieria systemów informatycznych od strategii do realizacji

Krzysztof Madrjas (IBM Polska)  
AD/Cycle IBM-owska filozofia inżynierii projektowania  
oprogramowania

**Od metodologii tworzenia systemów  
informatycznych do CASE  
(zamiast słowa wstępnego)**

Piąta Wiosenna Szkoła PTI zamyka cykl monotematycznych, tygodniowych, spotkań informatyków, na tematy związane z metodyką tworzenia systemów informatycznych. Majowe spotkania członków PTI mają już swoją tradycję i stałych bywalców.

Pierwsze majowe spotkanie w roku 1988 było wynikiem autentycznej potrzeby środowiska informatyków, które odczuwało potrzebę wymiany poglądów na temat metodyk tworzenia systemów informatycznych. W uporządkowaniu problemów metodycznych, upatrywano sposób na sprawniejsze i lepsze jakościowo wytwarzanie systemów informatycznych. W trakcie Pierwszej Wiosennej Szkoły PTI dokonano przeglądu poglądów różnych środowisk na problemy metodyk wytwarzania systemów informatycznych.

Druga Wiosenna Szkoła PTI skoncentrowała się na metodach, technikach i narzędziach tworzenia systemów informatycznych, dając przegląd stosowanych w praktyce wytwarzania systemów oraz informacje o nowinkach znanych z literatury. Niektóre referaty dawały nowe spojrzenie na znane od dawna i stosowane metody i techniki.

Trzecia Wiosenna Szkoła PTI wychodziła naprzeciw tendencjom, które były sygnalizowane w dwóch wcześniejszych spotkaniach. Referaty tej Szkoły dotyczyły komputerowego wspomaganie tworzenia systemów informatycznych. Obrady Szkoły koncentrowały się głównie wokół informacji o postępach i doświadczeniach w zakresie CASE w innych krajach. Jedno polskie wykorzystanie prostego pakietu, przedstawione w trakcie Szkoły, było jaskółką, która dopiero miała uczynić wiosnę. Wymiana poglądów była twórcza, chociaż wykazała duży dystans pomiędzy warsztatem pracy informatyka w Polsce a komputerowo wspomaganym wytwarzaniem stosowanych w innych krajach.

Kolejne edycje Wiosennej Szkoły PTI zawsze starały się poruszać problematykę, która aktualnie była najbardziej interesująca dla środowiska informatycznego, uzgadniając tematykę Szkół z kolegami organizatorami Górskich Szkół PTI.



Wydaje się, że działania te były skuteczne o czym może świadczyć frekwencja na kolejnych Szkołach.

Problemem początku lat 90-tych w Polsce stały się sieci mikrokomputerowe, które miały efektywnie spiąć moce licznych PC-tów, dając dodatkową wartość na miarę naszych możliwości. W referatach Czwartej Wiosennej Szkoły PTI przedstawiono problemy i rozwiązania w zakresie lokalnych sieci mikrokomputerowych. Zasygnalizowano problemy sieci rozległych, dzieląc się kilkuletnimi doświadczeniami zespołu, który pracował nad krajową siecią rozległą. Tematyka sieci a głównie doświadczenia z ich stosowania, obrazowały nowe warunki wytwarzania aplikacji.

Tematyka czterech Szkół przygotowała do podjęcia tematu, który od dawna był celem organizatorów. Chcieliśmy zaprezentować konkretne narzędzia wspomagające bardzo efektywnie, wytwarzanie systemów informatycznych. Prymitywne narzędzia wytwarzania stosowane w naszej praktyce powodowały frustracje informatyków i użytkowników tworzonych aplikacji. Wydaje się, że nadszedł czas zmian jakościowych, które umożliwią wykorzystanie wiedzy informatyków zgodnie z oczekiwaniami. Chcemy pokazać aktualne możliwości dostępne na naszym rynku i przedyskutować w fachowym gronie szanse i zagrożenia tej koniecznej zmiany jakościowej. O potrzebie takich działań świadczą liczne incjatywy i wypowiedzi, które mówią o systemach otwartych. System otwarty to pewna klasa produktu, na który istnieje zapotrzebowanie, my informatycy powinniśmy zastanawiać się, jak tworzyć takie systemy? Piąta Wiosenna Szkoła PTI powinna stanowić element tych rozważań.

Piąta Wiosenna Szkoła PTI spina tematykę dotychczasowych Szkół próbując przedstawić nowe podejście do wytwarzania systemów informatycznych. Nowe podejście metodyczne charakteryzuje się wykorzystaniem komputerowego wspomaganie w całym cyklu życia systemu. Powstało środowisko sprzętowo-programowe wspomagające wytwarzanie systemu, pozwalając skoncentrować się informatykom na rozwiązywaniu problemów a nie na zwalczaniu problemów technicznych czy technologicznych. Języki czwartej generacji, narzędzia CASE, systemy zarządzania bazami danych, możliwości i opcje systemów operacyjnych, to podstawowe elementy nowego środowiska, które wspomaga szybkie i sprawne wytwarzanie systemów o wysokiej jakości.

W pierwszym dniu gospodarzem będzie Digital Equipment Polska, sponsor Szkoły. W wystąpieniach swoich pracowników firma przedstawi nową koncepcję motodyczną tworzenia systemów informatycznych oraz środowisko CASE wspomagające wytwarzanie systemów - Digital's COHESION. Wykłady będą wspomagane pokazami narzędzi CASE na stacji roboczej produkcji firmy Digital. W uzupełnieniu przedstawiciel firmy CasePol, partner Digitala, przedstawi środowisko zintegrowane CASE, użytkowane na sprzęcie Digitala.

W drugim dniu gospodarzem będzie CSBI (Computer Systems for Business International LTD.), sponsor Szkoły. Wystąpienia pracowników CSBI będą koncentrowały się wokół języka PROGRESS oraz narzędzi typu CASE wywodzących się z tego języka lub współpracujących z tym językiem.

W trzecim dniu przewidujemy zajęcia praktyczne prowadzone przez zespół z Uniwersytetu Gdańskiego. Będzie to 1-dniowy tutorial na temat: Metody analizy i projektowania systemów w praktyce - szkolenie: wykłady, ćwiczenia, analiza przypadków, praca w grupach, dyskusje.

Czwarty dzień obsługiwany będzie przez sponsorów ZETO-RODAN i ASCOM. Przed południem przedstawiony zostanie przez dr Witolda Staniszkisa z ZETO-RODAN referat: "Inżynieria systemów informatycznych - od strategii do realizacji". Po południu firma ASCOM przedstawi swoją ofertę produktów oferowanych w Polsce.

Piąty dzień Szkoły obsługiwany będzie przez firmę IBM Polska, sponsora Szkoły, której pracownik wygłosi referat "AD/Cycle - IBM-owska filozofia inżynierii projektowania oprogramowania".

Pięć lat jakie minęły od Pierwszej Wiosennej Szkoły PTI w 1988 roku, to bardzo burzliwy okres w historii Polski. W tym czasie przechodziliśmy rewolucyjne zmiany jakościowe i ilościowe również w informatyce. Widać to porównując Pierwszą i Piątą Szkołę. Pierwsze Szkoły miały charakter akademicki, co oznacza że większość wykładowców reprezentowała środowisko akademickie a referaty najczęściej sumowały, oceniały, wytyczały kierunki zalecanych działań. W kolejnych edycjach zaczęły pojawiać się produkty handlowe, które oferowane uczestnikom pokazywały efektywne nowe sposoby wytwarzania systemów. Przedstawiona lista firm, prezenterów Piątej Wiosennej Szkoły wyraźnie tą tezę



potwierdza. Takie są realia obecnych czasów i konferencje dotychczas naukowe, muszą to uwzględniać.

Mimo, że staramy się zachować formułę organizacyjną Szkoły, to pewne zmiany organizacyjne są konieczne a wynikają z nie tylko naukowego charakteru Szkoły. Piąta Wiosenna Szkoła PTi będzie realizowana w dwóch współbieżnych warstwach: naukowo-merytorycznej i promocyjno-handlowej. Wprowadzenie komercyjnej warstwy w program Szkoły pozwoliło nam na znaczne obniżenie kosztów uczestnictwa. Mamy nadzieję, że innowację tą uczestnicy przyjmą ze zrozumieniem i pozwoli to na uatrakcyjnienie programu Szkoły. Staraliśmy się, żeby Firmy przygotowujące wystąpienia merytoryczne były równocześnie sponsorami Szkoły, co spowoduje, że wystąpienia promocyjno-handlowe będą w dużej części dotyczyły rozwiązań merytorycznie związanych z tematyką Szkoły.

Tradycyjnie, spodziewamy się ożywionej wymiany poglądów pomiędzy referentami a uczestnikami Szkoły. Życzę udanych obrad.

Zdzisław Szyjewski

Szczecin, kwiecień 1992 rok



# INFORMATYZACJA SUKCESEM BIZNESU

Wacław Iszkowski  
Digital Equipment Polska

Pewnie przynajmniej część czytelników pamięta książkę i hasło - "Informatyka kluczem do dobrobytu"<sup>1</sup>. Pozostawmy na boku weryfikację treści tej książki, pisanej w innej epoce i w innej sytuacji. Obecnie jednak sprawdza się główne hasło, gdyż właśnie teraz jedynie dzięki informatyce będzie możliwy dalszy rozwój gospodarczy. I co ważniejsze nie dotyczy to tylko Polski, ale przede wszystkim USA i reszty nowoczesnego świata. W tym eseju spróbujemy powiązać tę tezę z aktualnymi możliwościami informatyki<sup>1</sup>.

## Agresja Informatyki

Informatyzacja staje się ogólnosiwiatowym pojęciem, gdyż nawet w języku angielskim i amerykańskim pojawił się termin *informationalization*<sup>2</sup>. Dlaczego zajęto się hasłem ogólniejszym niż komputeryzacja? Dlaczego nawet Prezydent Bush dał przykład swoim rodakom, ucząc się obsługi komputera? Dlaczego pomimo coraz mniejszych zysków, czołowe firmy komputerowe inwestują ogromne pieniądze w badania nad nowymi technologiami i projektami systemów?

---

<sup>1</sup>) Targowski Andrzej; Informatyka Kluczem do dobrobytu; PWN 1970.

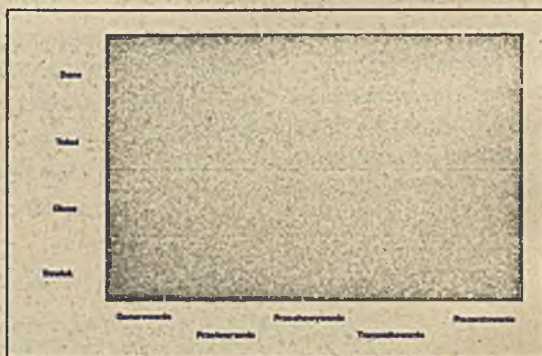
<sup>2</sup>) *Informationalization* jest nowym słowem (na wzór słowa *industrialization*) w języku amerykańskim (niektórzy wolą *infotize*, *infomate*, *informate*, *infotainment*, *infomediary*, *infocrate*) zbudowanym dla potrzeb opisu technologii przetwarzania informacji, któremu możemy spokojnie przypisać polskie słowo informatyzacja.

KOMPUTER	- to każdy wie co to jest,
KOMPUTERYZACJA	- zakup i użytkowanie komputera,
INFORMATYKA	- technologia przetwarzania informacji,
INFORMATYZACJA	- zakup i wykorzystywanie informatyki,

Na te wszystkie pytania można dać jedną odpowiedź. Informatyka osiągnęła wiek dojrzały i zażądała prawa decydowania o losach świata. Z drugiej strony gospodarka światowa w dotychczasowej formie staje się przestarzała. Era dominacji przemysłu z wszelkimi tego dobrymi i złymi skutkami przemija. Jedynie zmiana jakościowa ekonomii i organizacji gospodarki oraz społeczeństwa może pozytywnie wpłynąć na dalszy rozwój świata. Informatyka agresywnie zdobywa nowe pola wpływów i zaczyna dominować we wszystkich dziedzinach życia. Wielu ludzi, szczególnie starszych, fakt ten nieco przeraża.

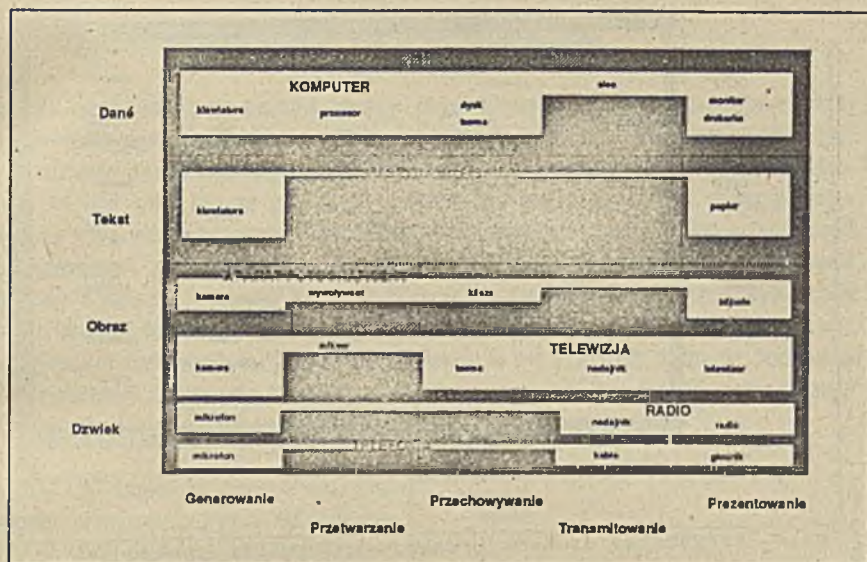
Zanim jednak spróbujemy przeanalizować ofertę informatyki - spójrzmy nieco na poprzednie lata. Człowiek w swoim środowisku posługuje się czterema formami przekazu informacji: dźwięk, obraz, pismo i inne dane, a w szczególności liczby (pozostałe zmysły człowieka - smak, dotyk i zapach będą być może wykorzystane do przekazywania informacji w przyszłości). Równocześnie każda z tych czterech form jest generowana, przetwarzana, przechowywana, transmitowana oraz prezentowana. Przedstawmy to w postaci tabeli (rys.1), w której wiersze identyfikowane są formami, a kolumny metodą obsługi informacji. A teraz wpiszmy w tabelę (rys.2) znane w początkach lat 60-tych urządzenia techniczne, starając się uwypuklić ich charakterystyczne cechy formy i obsługi informacji.

Na przykład radio - generowany dźwięk chwytyany jest przez mikrofon, a następnie jest nieco



Rys.1 Środki przekazu informacji

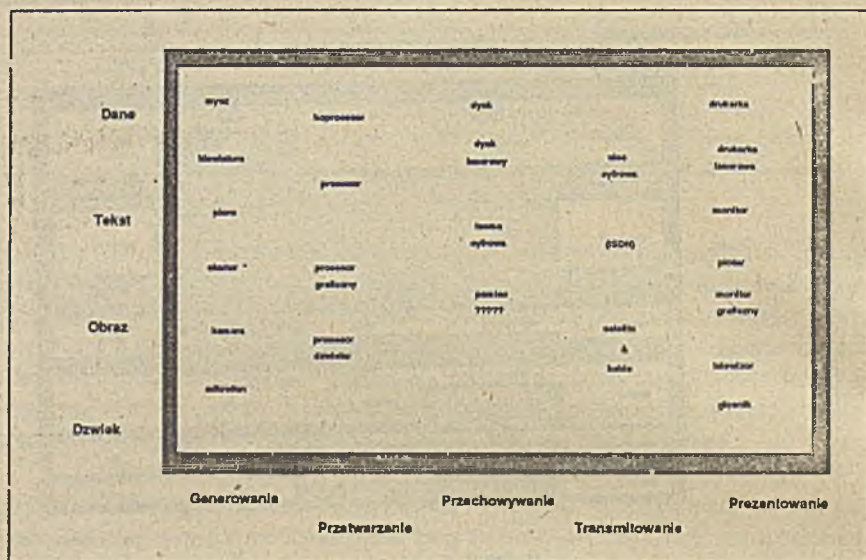




Rys.2 Przykłady środków przekazu informacji

przetwarzany (modulowany) i transmitowany z nadajnika na falach do odbiorników. Dodając urządzenia magnetofonowe możemy dźwięk przechowywać na taśmach. Inne urządzenie - telefon powstało w momencie wynalezienia sposobu przesyłania dźwięku drutem. Podobnie możemy przeanalizować takie urządzenia jak maszynę do pisania, aparat fotograficzny, telewizję i wreszcie komputery - przeznaczone do obliczeń. Proszę zauważyć, że każde z tych urządzeń oraz związane z nim infrastruktury były zarządzane przez różne firmy.

W kolejnych latach komputery zaczęły wchłaniać inne urządzenia techniczne, najpierw zaczęły przetwarzać teksty, a potem prezentować obraz na ekranach (najpierw 2D, potem 3D i wreszcie animowany). Rozwijając sieci komputerowe coraz więcej usług telekomunikacyjnych jest realizowane przez komputery (rys.3). A obecnie i w niedalekiej przyszłości wszystkie usługi będą realizowane przez system zwany informatycznym, gdyż komputer, a raczej procesor będzie tylko elementem tego systemu. Do generacji wykorzystywana będzie klawiatura (pewnie inna niż ta dzisiaj używana), mysz, mikrofon, skaner i płytka



Rys.3 Przyszłość środków przekazu informacji

uo odczytu pisma ręcznego. Komputer będzie się zajmował przetwarzaniem wszystkich form przekształconych w jednolitą formę cyfrową, a następnie ich mieszaniem. Do przechowywania będą wykorzystywane dyski laserowe i taśmy cyfrowe oraz inne pamięci. Do transmisji zostanie użyta sieć ISDN ze światłowodami docierająca do każdego mieszkania oraz ogólnoswiatowa sieć satelitarna i radiowa lokalna do połączenia ze sobą wszystkich nadajników i odbiorników. Do prezentacji zostaną wykorzystane telewizory cyfrowe nowych typów i inne mocno zmodernizowane urządzenia audiowizyjne jak na przykład wyświetlacze wielkości kartki papieru.

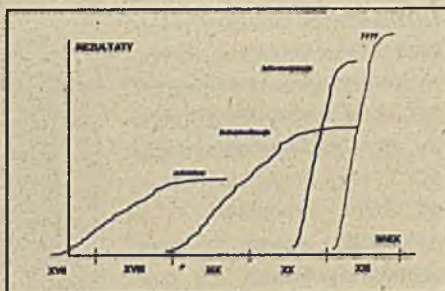
To nie żart, ale nawet pójście do ubikacji będzie obcowaniem z informatyką, która w tym wypadku przekaże do centrum medycznego dane o naszym zdrowiu, a poza tym dla uniknięcia straty czasu zapewni nam łączność ze światem. Jak można telefonować z samochodu, to dlaczego nie z ubikacji. Czy to nie jest agresja informatyki w nasz intymny mały świat oraz w całe społeczeństwo ?



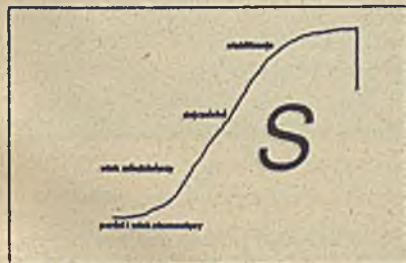
## Informatyzacja to coś więcej niż komputeryzacja

Dlaczego świat zgadza się na taką agresję, chociaż trzeba przyznać, że wzorem robotników protestujących przeciwko maszynom parowym, wielu nie chce tego akceptować, bo nie rozumie zasad funkcjonowania i boi się skutków wyrzucenia poza nawias społeczny. Istnieje tylko jedna przyczyna - ekonomia. Informatyka jako sama w sobie dostarcza tylko coraz mniejszych dochodów producentom sprzętu i oprogramowania i nieco większych projektantom systemów, a także konsultantom. Samo zakupienie komputera i jakiegoś systemu informatycznego nie przynosi kupującemu praktycznie żadnych korzyści finansowych, a tylko dużo kosztuje. W wielu instytucjach pracujący tam sprzęt jest już amortyzowany tylko na papierze, a nie przyniósł żadnych dochodów oprócz splendoru posiadania nowoczesnej techniki. A jednak to tylko informatyka może być katalizatorem wzrostu gospodarczego.

Cofnijmy się znowu w historię rozwoju gospodarczego świata (rys.4). W wiekach od XV do XVIII podstawą rozwoju gospodarczego Europy było rolnictwo. W wieku XVIII rozpoczęła się era industrializacji, której najlepsze lata wieku XIX i początku XX wieku przyniosły gwałtowny rozwój Europy i Ameryki. Krzywa S linii rozwoju



Rys.4 Ery rozwoju cywilizacji



Rys.5 Krzywa S

(rys.5) odpowiada cyklowi życia człowieka. Wi-  
dzimy na niej okres porodu, szybkiego wzrostu,  
okres wchodzenia w dojrzałość, a następnie  
przejście w zasłużony okres stabilizacji i trwa-  
nia aż do końca. Trzeba tutaj dodać, że cho-  
ciaż nie zanikło rolnictwo jako całość i długo  
nie zniknie typowa produkcja przemysłowa, to  
w całym tym okresie tysiące firm, podobnie jak  
wielu ludzi zniknęło z tego świata. Z listy Fortune 500 z roku 1955 ponad 60% firm dzisiaj

nie istnieje. Żadna, nawet najlepiej zorganizowana, firma nie może dzisiaj żyć w świadomości, że będzie istnieć wiecznie. Sprzedaż, likwidacja czy bankructwo jest wpisane w historię każdej firmy w chwili jej powstania.

Pierwsze komputery powstały w latach 40-tych. Początek ery informatycznej nastąpił w roku 50-tym. Przez wiele następnych lat wpływ informatyki na życie społeczeństwa i rozwój gospodarczy był niewielki. Budowano coraz kosztowniejsze maszyny, pisano coraz dłuższe programy, komputeryzowano banki i księgowość i wreszcie powstały mikrokomputery, które weszły pod strzechy. Lata 90-te są początkiem okresu dojrzałości informatyki i jej gwałtownego wpływu na rozwój gospodarczy świata, ale pojedynczo tylko tych firm, które zrozumieją jak zastosowanie informatyki zamienić w zyski. Era informatyzacji ma trwać tylko 70 lat do roku 2020 (to już za 28 lat!).

#### Historia ery industrializacji

- 1733 - latające czółenko podwoło prędkość tkania
- 1765 - maszyna dziewiarska zamiast 100 przadek
- 1784 - maszyna parowa Jamesa Watta
- 1784 - szybki rozwój przemysłu w Anglii i Płn.Stanach
- 1807 - pierwszy statek parowy
- 1825 - pierwsza na świecie linia kolejowa w Anglii
- 1839 - telegraf igłkowy
- 1848 - przyspieszenie rozwoju przemysłu w Europie
- 1878 - organizacja badań - T.Edison
- 1880 - patent na żarówkę
- 1889 - przemysłowa wystawa światowa w Paryżu
- 1901 - fale radiowe pokonują Atlantyk
- 1903 - powstanie samolotu
- 1905 - seryjna produkcja samochodów w Detroit
- 1908 - Ford Model T - samochód powszechny
- 1913 - montaż taśmowy samochodu
- 1919 - przelot nad Atlantykiem
- 1919 - przekształcenie jądra atomu
- 1923 - 2 mln Fordów rocznie
- 1927 - popularność turystyki kolejowej
- 1935 - parowiec Normandie pokonał Atlantyk
- 1937 - otwarcie Golden Gate
- 1937 - pierwszy odbiornik telewizyjny
- 1940 - pierwszy komputer
- 1945 - pierwszy tranzystor
- 1945 - próba bomby atomowej
- 1950 - telewizor kolorowy
- 1951 - pierwszy wideorekorder
- 1952 - przekroczenie bariery dźwięku
- 1953 - stworzenie modelu budowy DNA
- 1954 - podwodna łódź atomowa
- 1954 - pierwsza elektrownia jądrowa
- 1957 - sputnik na orbicie
- 1961 - pierwszy człowiek w kosmosie
- 1969 - pierwszy człowiek na Księżycu

Powstanie sieci informatycznych i dostęp do systemu z każdego prawie miejsca pobytu jest właśnie tym kluczem do uzyskania zysków i zwiększenia dochodów firmy. Produkto-

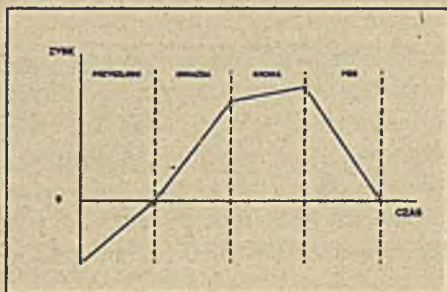


wanie tylko na zamówienie konkretnego klienta (nawet chusteczki do nosa) przy minimalizacji stanu zapasów, zautomatyzowanie samego procesu projektowania wyrobu i jego produkcji, przeniesienie części pracowników do ich domów z podłączeniem terminala, dostęp do informacji i możliwość jej sprawnego zagregowania i przeanalizowania, itd. itd. To są tylko ogólne przykłady, gdy tylko informatyzacja może przynieść dochód - oczywiście zastosowana i wykorzystywana poprawnie.

A jak jest w Polsce ? Szczęśliwie udało się nam uczestniczyć w procesie fascynacji mikrokomputerami (trochę na prawach złodzieja). Młode pokolenie traktuje te pecety już naturalnie, dla starszych są one nieco zbyt skomplikowane. Przez to jednak uważane są za szczyt nowoczesności. W gospodarce i administracji realizowany był proces komputeryzacji, polegający na zakupie peceta, dla którego potem wymyślano odpowiedzialne zadania. Nas w Polsce, nawet gdy nie ma pieniędzy, nawet gdybyśmy nie chcieli związać się gospodarczo z Europą czeka proces informatyzacji. Proces ten może być dłuższy niż w Japonii, USA czy w Europie, ale bez niego po prostu zginiemy. I co gorsza już bardzo dużo tracimy na niesprawnym systemie bankowo-finansowym, na braku rzeczywistych informacji o stanie gospodarki państwa, na niekonkurencyjności produkcji spowodowanej brakiem odpowiedniej technologii projektowania i nadzoru technicznego. Co ciekawsze, wdrożenie kolejnego systemu finansowo-księgowego prawie nie daje zysku.

## Gdzie jest biznes ?

Każda firma produkując jakieś wyroby, lub dostarczając jakichś usług dokonuje analizy ekonomicznej porównując się z konkurencją i obserwując swoje zyski i oczywiście starając się myśleć o przyszłości. Obrazem takiej analizy może być kwadrat bostoński (tab.1) (opracowany przez Boston Consulting Group w latach 1960. .1970) oraz jego rozwinięcie w postaci wykresu (rys.6). Lewa dolna część kwadratu - *dojna KROWA* oznacza uzys-



Rys.6 Zyski z produkcji Cosia

Tabela I. Kwadrat Bostoński

Ren- tow- ność > 10%	<i>wschodząca</i> <b>GWIAZDA</b>	<b>PRZYSZŁOŚĆ</b>
10% < Ren- tow- ność	<i>dojna</i> <b>KROWA</b>	<i>zdychający</i> <b>PIES</b>
	pozycja na rynku >50%> pozycja na rynku	

kanie danym produktem ponad 50% rynku przy stosunkowo niskim wskaźniku rentowności, czyli duży obrót - mały zysk jednostkowy. Prawa dolna część kwadratu - *zdychający* **PIES** oznacza upadającą wartość oferowanego produktu - duża konkurencja wypiera nas z rynku, a szanse wzrostu są niewielkie. Górna część kwadratu oznacza przyszłość. Po lewej stronie mamy *wschodzącą* **GWIAZDĘ**, czyli produkt będący "hitem" na rynku. Oczywiście każdy taki produkt staje się z czasem *dojną* **KROWĄ**, a potem *zdychającym* **PSEM**. Biznes jest w tym, aby stało się to jak najpóźniej, po zwrocie poniesionych nakładów. Prawa górna część kwadratu jest naszą **PRZYSZŁOŚCIĄ**. Jeżeli nie mamy tutaj nic do zaoferowania to szanse przeżycia naszej firmy są małe.

Teraz możemy już zacząć myśleć, jak przy pomocy informatyki możemy pomóc naszej firmie. Oczywiście zakupy systemów informatycznych - aplikacji powinny nastąpić już teraz natychmiast dla *dojnej krowy*, być może też dla podtrzymania upadającej produkcji. Na pewno należy zaplanować odpowiednie aplikacje dla *wschodzącej gwiazdy* i wna następne lata dla *przyszłości*. Dodatkowo dla każdej z potrzebnych aplikacji musimy się zastanowić, czy jej istnienie jest WITALNE (niezbędne), czy tylko WSPOMAGAJĄCE, albo czy ma wkrótce STRATEGICZNE znaczenie lub w nieco dalszej PRZYSZŁOŚCI. Proste,



prawda - tylko jak uzyskać taką analizę ? Przykład efektu takiej analizy jest pokazany w kwadracie bostońskim (tab.II).

Tabela II.: Przykładowe aplikacje dla wspomagania fazy biznesu

STRATEGICZNE	PRZYSZŁOŚCIOWE
<ul style="list-style-type: none"><li>* Przygotowanie produkcji</li><li>* Planowanie produkcji</li><li>* Powiązania z dostawcami</li><li>* Kontrola jakości</li><li>* Prognozowanie sprzedaży</li><li>* Analiza rentowności</li></ul>	<ul style="list-style-type: none"><li>* Wymiana z dostawcami</li><li>* Planowanie zatrudnienia</li><li>* Poczta elektroniczna</li><li>* Wspomaganie decyzji</li><li>* Systemy ekspertowe</li><li>* Przetwarzanie dokumentów</li></ul>
<ul style="list-style-type: none"><li>* Baza danych pracowników</li><li>* Planowanie remontów</li><li>* Gospodarka magazynem</li><li>* Kontrola sprzedaży</li><li>* Projektowanie produktów</li><li>* Baza danych półproduktów</li><li>* Rozliczenia faktur</li></ul>	<ul style="list-style-type: none"><li>* Kontrola czasu pracy</li><li>* Kontrola budżetu</li><li>* Raportowanie wydatków</li><li>* Analiza kosztów</li><li>* Księgowość i finanse</li><li>* Płatności</li><li>* Przetwarzanie tekstów</li></ul>
WITALNE	WSPOMAGAJĄCE

---

Informatyzowanie firmy nie może być tylko procesem technicznym znalezienia, lub napisania odpowiedniej aplikacji i zainstalowania jej łącznie z komputerem. Wprowadzając informatykę do firmy nie można popełnić błędów fabrykantów z początku XIX wieku. Systemy informatyczne tak bardzo wpływają na zmianę zachowań użytkowników ,że bez

---

**Informatyka dodana do bałaganu  
tworzy bałagan do kwadratu.**

---

zmian w świadomości pracowników nie można oczekiwać zakładanych efektów. Kto i w jaki sposób może dokonać odpowiedniej analizy organizacji zarządzania firmą oraz zespołem ludzi, a następnie zaproponować sensowne rozwiązanie porządkujące ?

## Realizacja systemu informatyzacyjnego

Zanim odpowiemy na te pytania, zbudujemy model życia systemu informatyzacyjnego, włączając doń wnioski z ostatnich rozważań.

SYSTEM INFORMATYZACYJNY  
jest systemem obejmującym  
funkcjonowanie ekonomiczne i organizacyjne firmy  
odzworowane w aplikacji  
systemu informatycznego.

W modelu cyklu życia takiego systemu związane są ze sobą rzeczy z czynnościami, które muszą być dla nich wykonane. W naszym modelu wyróżniamy dwie grupy rzeczy - BIZNES i INFORMACJE. W prowadzeniu BIZNESU firmy mamy cele (*objectives*) jej rozwoju oraz procesy zachodzące wewnątrz i zewnątrz firmy. W realizacji przetwarzania INFORMACJI mamy przepływy (*flows*) informacji w aplikacjach oraz systemy wykonujące zrealizowane aplikacje. Po stronie czynności wyróżniamy PLANOWANIE przedsięwzięć z opracowaniem koncepcji oraz analizy, a następnie PROJEKTOWANIE rozwiązania i REALIZACJĘ aplikacji systemu informatycznego z wykonaniem Implementacji i Instalacji. Nad prawidłowym przebiegiem tych prac czuwa ZARZĄDZANIE.

W modelu realizacji informatyzacji firmy znajdujemy wykaz czynności jakie powinny być wykonane, aby móc przygotować firmę, zaprojektować, zrealizować oraz eksploatować system informatyczny w firmie. Podany model, w którym można dyskutować stopień podziału na poszczególne fazy życia, przedstawia aktualną nowoczesną metodę informatyzacji firmy. Jest to jedyna prawidłowa droga - "OD ANALIZY CELÓW BIZNESU DO INSTALACJI KOMPUTERA", zamiast dotychczas najczęściej stosowanej komputeryzacji - "OD KUPNA KOMPUTERA DO POSZUKIWANIA DLA NIEGO PRACY".

Teraz możemy wstępnie wskazać metody wdrażania informatyki w organizację firmy. Przede wszystkim konieczne jest bardzo wczesne przeanalizowanie potrzeb i możliwości firmy. Co więcej analizy te muszą być wykonane przez zewnętrzne w stosunku do firmy zespoły konsultantów przy ściślejszej współpracy z dyrekcją. Równocześnie zespół konsultan-



Tabela III. Model realizacji informatyzacji firmy

		PLANOWANIE		PROJEKTO- WANIE	REALIZACJA		ZARZĄ- DZANIE
		Kon- cepcja	Ana- liza	Projekt	Imple- mentacja	Insta- lacja	Operacja
B U Z L N E S	Cele	Konsultacje w zarządzaniu firmą dla określenia jej celów  Strategie wyko- rzystania in- formacji	Ewaluacja problemów Analiza potrzeb				
	Pro- cesy	Makro-analiza procesów prowa- dzenia biznesu	Miary pro- cesów prze- pływu in- formacji				
I N F O R M A C J A	Prze- pływy	Mikro-analiza przepływu in- formacji w firmie	Zaprojekto- wanie lub przys- tosowanie aplikacji	Implementacja, testowanie i uru- chomienie aplika- cji			
		Konsultowanie wykorzystania gotowych apli- kacji		Zarządzanie realizacją pro- jektu			
	Sys- tamy	Serwis wspo- magania zakupu systemu	Projekt konfigura- cji i in- tegracji sieci	Konsultacje tech- niczne w instala- cji i testowaniu systemu	Szkolenia perso- nelu i użytkow- ników	Serwis i pieleg- nacja systemu	
		Przygotowanie zapytania of- ertowego	Konfigura- cja plat- formy				
			Produkty sprzętowe i oprogramowania - wybór, zakup i dostawa i instalacja eksploatacyjna				

tów musi mieć odpowiednią wiedzę w zakresie ekonomii, menedżerstwa, profilu analizowanego biznesu oraz w możliwościach informatyki, czyli znajomości istniejących aplikacji, oprogramowania i sprzętu oraz narzędzi wspomagających.

## Inżynieria Informatyzacji

W ostatnich latach - w Polsce jeszcze dość nieśmiało - pojawiała się nowa zawodowa specjalność informatyczna - inżynieria oprogramowania (*software engineering*). Zakresem jej działania jest opracowywanie metodyki i technologii projektowania i realizacji systemów informatycznych, czyli aplikacji oraz oprogramowania systemowego. Efektem działalności tej grupy są narzędzia tworzące komputerowo wspomagane środowisko oprogramowania (*CASE - Computer Aided Software Environment*). Obecnie jednak należy się zastanowić nad powstaniem nowej specjalności - inżynierii systemów (*system engineering*) i równocześnie nieco zmienić znaczenie pojęcia CASE (... *Systems...*) na komputerowo wspomagane środowisko systemów.

Co więcej, akceptując dotychczasowe rozważania o roli informatyzacji, warto już powołać i określić jej zadania, nową specjalność - inżynierię informatyzacji (*informationalization engineering*). Zadaniem tych specjalistów byłaby integracja systemu informatyzacyjnego poprzez zaprojektowanie i zrealizowanie wszystkich faz prowadzących do uzyskania gotowego systemu, złożonego z aplikacji przygotowanych na podstawie szczegółowych analiz celów i potrzeb firmy, uzasadnionych ekonomicznie. Oczywiście tacy specjaliści już istnieją (jest ich co prawda niewiele) oraz znane jest już wiele prac teoretycznych i metodycznych jak się zabrać do tego dość złożonego problemu.

Obecnie elementy fazy PLANOWANIA są realizowane głównie poprzez konsultacje i sesje analityczne, zorganizowane według precyzyjnie przygotowanych scenariuszy. Stosowane są tutaj analizy mocy i słabości oraz możliwości i ograniczeń (*SWOT - Strengthes, Weaknesses, Opportunities, Threats*) funkcjonowania danej jednostki biznesu. Wykorzystuje się również metody aktywnych dyskusji i burz mózgów. Często efektem takich sesji są proste w formie i czytelne dla uczestników plansze i schematy. Wartość tego typu konsultacji jest zależna od efektywnego udziału i zrozumienia problemu przez dyrekcję firmy.<sup>3</sup>

---

<sup>3)</sup> Koszt konsultacji kształtuje się w wysokości od 1500 do 4000 dolarów za dzień pracy konsultanta bez kosztów jego podróży i zakwaterowania oraz samych kosztów organizacji szkolenia.



## Narzędzia wspomagania Informatyzacji

Inżynieria informatyzacji potrzebuje również wspomagania komputerem czyli specjalnych narzędzi do analizy celów i potrzeb biznesu, a także możliwości wykorzystania informatyki dla zwiększenia efektywności finansowej firmy. Obecnie możemy jedynie odnotować pierwsze próby tworzenia takiego oprogramowania bazującego na systemach ekspertowych. Większość tych narzędzi będzie początkowo wspomagać zbieranie i porządkowanie zbieranych informacji. Z czasem możliwe będzie opracowanie standardowych analiz dla typowych firm i działalności. Rozprzestrzenienie się systemów komputerowych z nowoczesnymi terminalami z dostępem do zintegrowanych systemów informacyjnych pozwoli na łatwiejszą analizę porównawczą stanu biznesu firmy.

Teraz możemy oszacować (tab.IV) stopień istnienia, w stosunku do zapotrzebowania narzędzi wspomagania dla inżynierii informatyzacji, w tym również dla inżynierii systemów i oprogramowania (jest to bardzo szacunkowa subiektywna ocena autora). Wnioski wynikające z analizy tej tabeli, po pierwsze poddaję pod dyskusję czytelników z oczekiwaniem na ich uzupełnienie, a po drugie bez względu na wynik tej dyskusji możemy podać jeden wniosek bezdyskusyjny podany w ramce.

W inżynierii informatyzacji jest jeszcze bardzo dużo do zrobienia dla uzyskania efektywnych narzędzi wspomagających analizę, projektowanie i realizację, a następnie pielęgnację złożonych systemów informatyzacyjnych.

Tabela IV. Wspomaganie inżynierii informatyzacji.

	ZADANIA	WSPOMAGANIE	CASE
1	Konsultacje w zarządzaniu firmą (TOP-MAPPING, RANS)	systemy ekspertowe	5 %
2	Strategie wykorzystywania informacji (DISS)	bazy danych	5 %
3	Makro-analiza procesów prowadzenia biznesu	programy graficzne	5 %
4	Ewaluacja problemów	dokumentacja	10 %
5	Miary procesów przepływu informacji (met. YOURDONA, etc)	analiza grafów	15 %
6	Mikro-analiza przepływu informacji (grafy GANTTA)	rysowanie i analiza grafów	20 %
7	Konsultowanie aplikacji	bazy danych aplikacji	10 %
8	Projekt aplikacji	projektowanie, słowniki	50 %
9	Implementacja i testowanie	generatory, debugery	90 %
10	Zarządzanie realizacją projektu	systemy planowania i nadzoru zadań	80 %
11	Serwis wspomaganie zakupu systemu	systemy analiz kosztów	60 %
12	Przygotowanie zapytania ofertowego	budowanie dokumentacji	70 %
13	Projekt konfiguracji	systemy projektowania sieci	60 %
14	Konfiguracja platformy	systemy weryfikacji konfiguracji	30 %
15	Produkty sprzętowe i programowe	"problem dostawcy"	
16	Konsultacje techniczne	systemy - multimedia	20 %
17	Szkolenia	systemy edukacyjne	20 %
18	Instalacja	"problem dostawcy"	
19	Serwis produktów	"problem dostawcy"	
20	Analiza realizacji systemu informatyzacyjnego	systemy gromadzenia dokumentacji	10 %



## Zamiast podsumowania

Zaprezentowanego materiału nie można podsumować. Jest to bowiem jeszcze nie uporządkowane spojrzenie na to co nas czeka, a więc jest to w pewnym stopniu futurolo-

**DIGITAL WIZARDS**

Computers, phones and consumer-electronics goodies are coming together to create an explosion of new products and services that could change all our lives and soon. From smart TVs to machines that carry whole encyclopedias in a briefcase, the new gadgets might be a chance for American companies to regain a share of the consumer-electronics industry.

Super TV	Organizer	Communicator
<p>Your television could become a window on a new world of shows and services with a clear digital picture.</p> <ul style="list-style-type: none"><li>* Users bank, pay bills and shop via TV</li><li>* Video libraries bring movies on demand</li><li>* On-board computer find shows, so's VCR</li><li>* Early models: 1993</li></ul> <p>A video store at home Shop on screen</p>	<p>These silicon secretaries could replace the notes and phone slips that clutter your appointment diary and address book.</p> <ul style="list-style-type: none"><li>* Alerts user to appointments</li><li>* Stores phone numbers and notes</li><li>* Some models could take voice commands</li><li>* Early versions available</li></ul> <p>Voice activated! It's a Rolodex! Meeting time!</p>	<p>Part phone, part pager, it will send and receive messages, keeping you in touch with the office wherever you roam.</p> <ul style="list-style-type: none"><li>* Radio links to voice mail, electronic mail</li><li>* Remote access to office computer</li><li>* Electronic pen could eliminate keyboards</li><li>* Early versions: 1992</li></ul> <p>Voice activated! Call you computer! Send faxes!</p>

from Newsweek, April 6, 1992

gia. Jaką mamy gwarancję, że spełnią się założenia totalnej informatyzacji gospodarki i społeczeństwa. Jest to bardzo bliska przyszłość i chyba nam grozi. A co nas, lub nasze dzieci czeka po roku 2020? Tutaj jest już trochę trudniej stwierdzić jaką erą nastąpi

potem. Ja w każdym razie mogę opowiedzieć następującą historyjkę.

Kiedyś mój 14-letni syn stwierdził, że nie chce się zajmować informatyką. Na pytanie - a czym chciałbyś się zajmować? - odpowiedział - *chcę się zajmować biologią*. Wtedy wyjaśniłem mu, że to będzie właśnie przyszłość nowoczesnego świata i komputery będą bioniczne. Nieco się załamał tą perspektywą, że jednak będzie to bioputer, albo coś podobnego.

---

Artykuł ten został opracowany dla kwartalnika DECforum i za zgodą Redakcji z Digital Equipment Polska jest publikowany w materiałach konferencji PTI - CASE'92 .

© Digital Equipment Polska  
Wszelkie prawa zastrzeżone.



# *Środowisko COHESION*

---



Digital Equipment Polska

COHESION, 1

## *Oprogramowanie*

---

### ☐ Czym jest ?

- Fizyczny produkt, który może zostać uruchomiony na komputerze i wykonujący określone zadania
- Logiczna jednostka zbudowana na podstawie pewnego projektu

### ☐ Charakterystyka

- Tworzone na zamówienie (klienta, rynku)
- W większości budowane od podstaw (coś więcej niż zebranie odpowiednich kostek produktów razem)
- Stosowane w bardzo szerokim zakresie (nigdy do końca nie można przewidzieć jak i gdzie będzie stosowane)



## *Inżynieria oprogramowania*

---

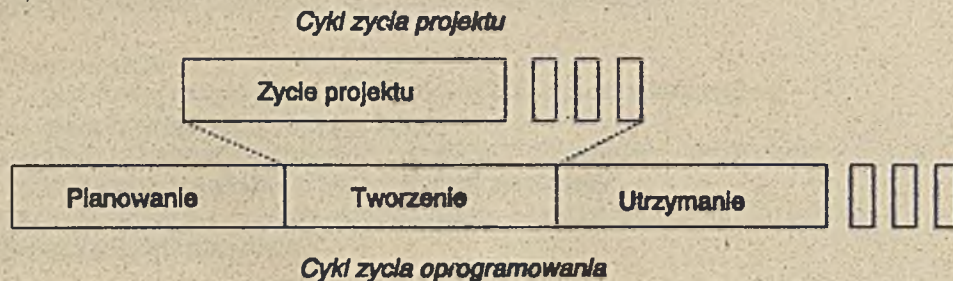
Zadaniem inżynierii oprogramowania jest projektowanie i tworzenie oprogramowania.

### ☐ Charakterystyka

- Wynikiem jest system o dużym stopniu złożoności
- Praca ma charakter zespołowy
- Duża ilość błędów w produkcji
- Większość działań może być wykonana przy użyciu komputerów

## Cykl życia oprogramowania

Oprogramowanie rodzi się i umiera. Każdy kawałek programu ma swój moment narodzin i śmierci, nic nie żyje wiecznie.





## Modele cyklu życia projektu

Klasyczny model cyklu życia projektu



Strukturalny model cyklu życia projektu



Cykl życia projektu z prototypem



## ***Modele cyklu życia projektu - c.d.***

---

### ☐ **Cykl życia projektu strukturalnego**

- Większość czynności może być wykonywana równolegle
- Zakończenie fazy nie jest obligatoryjne, zawsze można dokonać nawrotu w przypadku wystąpienia błędu lub zmian w koncepcji

### ☐ **Cykl życia projektu z prototypowaniem**

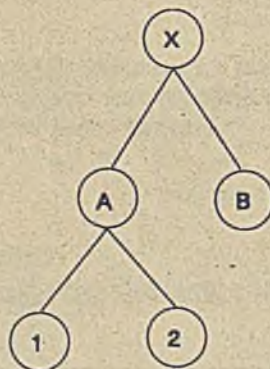
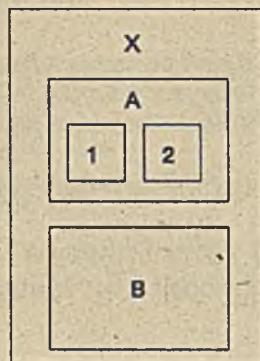
- Prototyp nie tylko całości ale także części systemu
- Możliwość zlekceważenia wymagań czasowych i efektywnościowych
- Dopuszczalne pozostawienie błędów w działającym prototypie



## *Modularność*

---

Podział systemu na logiczne części, które razem tworzą system. Moduły muszą posiadać bardzo dobrze zdefiniowane łącze z innymi modułami. Łączność może mieć wiele fizycznych postaci, reprezentowanych za pomocą pewnych abstrakcji (np. podejście obiektowo-zorientowane)



## *Strategie*

---

Generalnie istnieją dwa podejścia (strategie) dotyczące kolejności implementacji kolejnych części projektu.

☐ **Z dołu do góry (klasyczny)**

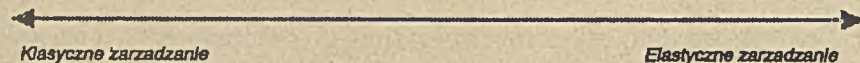
- Rezultat ostateczny otrzymujemy na końcu projektu
- Łatwiej wyklucza się błędy w modułach, trudniej błędy projektu
- Zapotrzebowanie na zasoby komputera rośnie wraz z rozwojem projektu

☐ **Z góry do dołu**

- Wersja systemu, która coś robi powstaje we wczesnej fazie projektu
- Błędy projektowania i analizy są wykrywane wcześniej
- Zapotrzebowanie na zasoby jest zbalansowane w czasie



# Zarządzanie projektem



## Różnice

Sekwencyjne przetwarzanie  
Implementacja po projekcie  
Z dołu do góry  
Sztywne reguły postępowania

Równoległe przetwarzanie  
Implementacja z projektem  
Z góry do dołu  
Dowolne reguły postępowania

## Niebezpieczeństwa



## ***Cykl życia projektu***

---

- ☐ **Czynności wstępne**
  - Analiza rynku
  - Plan ekonomiczny
- ☐ **Analiza systemu**
  - Formalne techniki
- ☐ **Projekt systemu**
  - Projekt logiczny (ogólny)
  - Projekt fizyczny (szczegółowy)
- ☐ **Kodowanie**
  - Dokumentowanie kodu
- ☐ **Testowanie**
  - Modułów
  - Łączy pomiędzy modułami
- ☐ **Utrzymanie oprogramowania**



## *Narzędzia CASE*

---

Narzędzia CASE ułatwiają czynności inżynierskie poprzez użycie zasobów komputera w miejsce zasobów ludzkich.

### ☐ Upper CASE i Lower CASE

- Upper CASE - analiza i projekt
- Lower CASE - kodowanie, testowanie i utrzymanie

### ☐ Definicja firmy Digital

Każde narzędzie użyte w procesie tworzenia oprogramowania jest narzędziem CASE

## ***Dlaczego narzędzia CASE ?***

---

- ☐ **Automatyzacja czasochłonnych czynności**
- ☐ **Formalne metody analizy i projektowania**
  - Modułowość
  - Badanie spójności i poprawności projektu
- ☐ **Możliwość współpracy wielu osób jednocześnie**
- ☐ **Klient łatwiej rozumie projekt i plan pracy**
- ☐ **Możliwość ponownego użycia istniejącego oprogramowania**



## ***Wymagania stawiane narzędziom CASE***

---

- ☐ **Przyjemność, łatwość i efektywność użycia**
- ☐ **Pokrycie całego cyklu życia oprogramowania**
- ☐ **Integracja narzędzi CASE**
  - Wspólny interfejs
  - Wspólny dane
  - Współdziałanie

## ***Integracja narzędzi CASE - Składnica informacji***

---

Rozwiązaniem problemu integracji narzędzi CASE jest technika Składnicy informacji. Technika ta oznacza, iż wszystkie narzędzia dzielą całą dostępną informację we wspólnej bazie danych zwanej Składnicą. Baza taka powinna być widziana jako obiektowo-zorientowana i powinna posiadać standardowy interfejs.

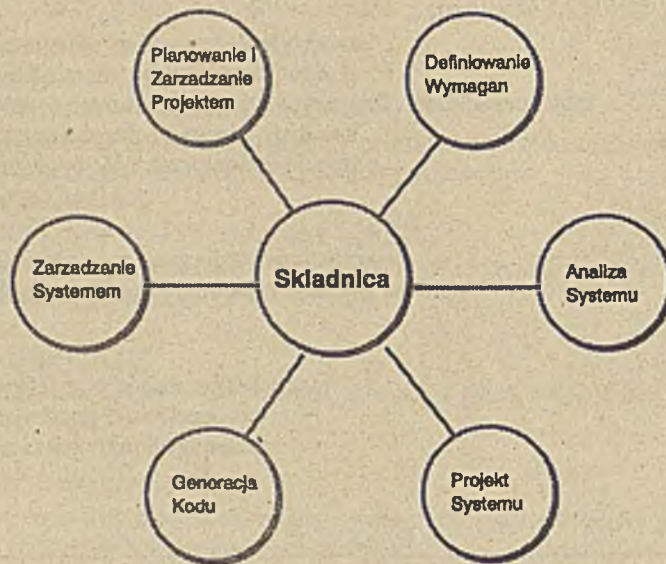
### **□ Zalety**

- Różne narzędzia różnych producentów mogą zostać zintegrowane
- Różnica pomiędzy różnymi środowiskami przestaje mieć znaczenie
- Użytkownicy mogą tworzyć własne środowiska



## *Składnica informacji w cyklu życia projektu*

---



## *Środowisko COHESION*

---

### ☐ Zakres

- Tworzenie oprogramowania
- Rozwijanie oprogramowania
- Zarządzanie oprogramowaniem

### ☐ Cel

- Wysoka jakość tworzonego oprogramowania
- Szybkość tworzenia oprogramowania

### ☐ Charakterystyka

- Wszystkie typy tworzonych aplikacji
- Heterogeniczna platforma sieciowa
- Różne platformy docelowe tworzonego oprogramowania
- Zintegrowana Składnica informacji
- Konsultacje, szkolenia i współpraca



## *Elementy środowiska COHESION*

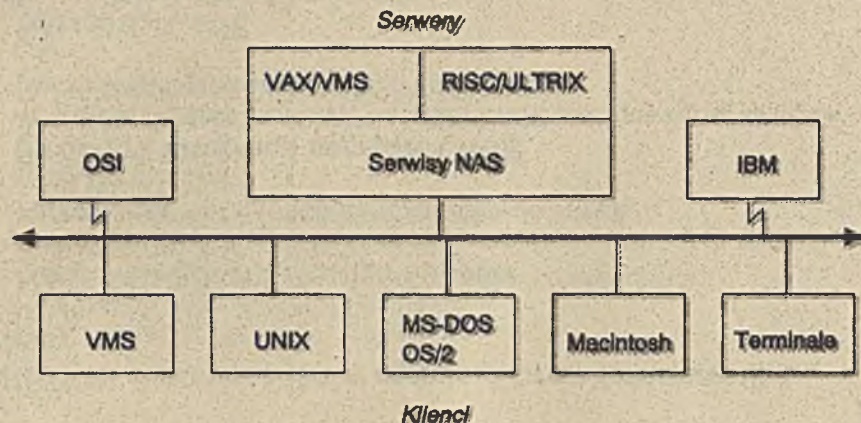
---

COHESION zbudowane jest w oparciu o cztery fundamentalne elementy.

- ☐ **NAS Network Application Support**  
Architektura NAS umożliwia tworzenie przenośnych i otwartych aplikacji w rozproszonym, heterogenicznym środowisku sieciowym.
- ☐ **Struktura integracji narzędzi CASE**  
W oparciu o uznane standardy, narzędzia CASE zintegrowane są na trzech płaszczyznach: prezentacji, kontroli i danych.
- ☐ **Narzędzia CASE**  
Narzędzia firmy Digital i innych firm pokrywają pełny cykl życia oprogramowania.
- ☐ **Serwisy i wsparcie**  
Digital oferuje konsultacje, szkolenia i wsparcie na różnych poziomach i etapach (pracy) tworzenia oprogramowania.

## ***COHESION w środowisku heterogenicznym***

Środowisko COHESION wspomaga tworzenie aplikacji, które będą działały na maszynach innych producentów. Kluczem do tego jest architektura NAS.





## *Narzędzia COHESION i NAS*

---

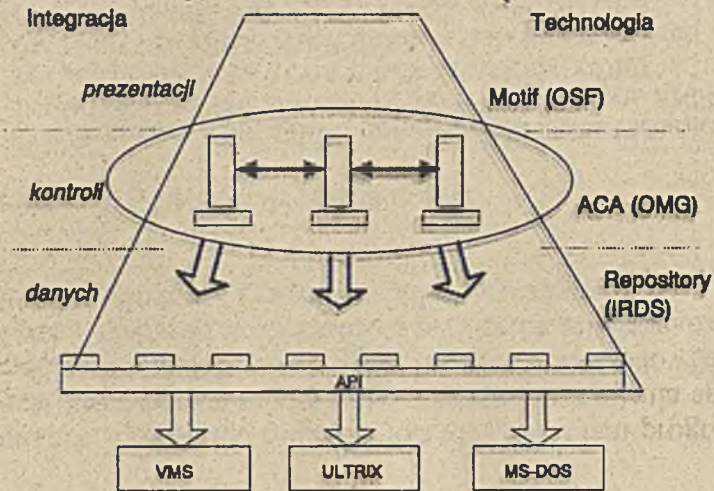
NAS (Network Application Support) jest ideą i zbiorem programów usługowych, które tworzą model projektowania i rozwijania aplikacji działających w środowisku sieciowym. Istotną cechą środowiska NAS jest jego pełna zgodność z większością przyjętych międzynarodowych standardów związanych z systemami informatycznymi.

### ☐ Komponenty COHESION oparte o NAS

- CDD/Repository - ANSI IRDS (Information Resource Dictionary Services)
- DECWindows Motif - X Windows, Motif (OSF, X/Open, ANSI)
- DECdesign, DECwrite - CDA (zawiera PostScript, ODIF)
- DECforms - ISO FIMS (Form Interface Management System)
- DEC PHIGS, DEC GKS - ANSI PHIGS, ANSI/ISO GKS

## Struktura integracji narzędzi CASE

Poszczególne narzędzia CASE to za mało. Niezbędna jest integracja tych narzędzi.





## *Struktura integracji w środowisku COHESION*

---

- ☐ **Integracja na poziomie prezentacji**  
Odnosi się do wspólnego interfejsu użytkowego wszystkich narzędzi i aplikacji (DECwindows Motif).
- ☐ **Integracja na poziomie danych**  
Pozwala aplikacjom na dzielenie informacji poprzez cały cykl życia oprogramowania (standard ATIS - A Tools Interface Standard).  
ATIS został stworzony przez Digital i Atherton Technology of Sunnyvale.  
ATIS jest obecnie ewaluowany przez CASE Integration Service Standards Committee. ATIS jest ewaluowany przez komitety ANSI X3H4 IRDS i ISO IRDS.
- ☐ **Integracja na poziomie kontroli - współdziałanie aplikacji**  
Udostępnia serwisy, pozwalające na współdziałanie, kontrolę i komunikację pomiędzy aplikacjami (standard ACA - Application Control Architecture).  
ACA został pierwotnie zaimplementowany jako mechanizm LiveLink w CDA. Obecnie, jego rozszerzenie zostało zaproponowane jako podstawa w standardzie Object Request Broker - Object Management Group.

## ***Składnica informacji***

---

Składnica jest rozwinięciem idei słownika, posiadającego właściwości integracji danych. Składnica informacji rozwija tą ideę dodając serwisy zapewniające bezpieczeństwo, zgodność wersji i zarządzanie obiektami.

### ☐ **Cechy składnicy informacji**

- Dostęp do definicji danych (rekordy, bazy, formularze, procedury, reguły)
- Informacje o obiektach (moduły kodu, zbiory, modele i dokumenty)
- Informacje o metodach (operacje na obiektach, tworzenie, edycja)
- Dzielenie obiektów i zarządzanie nimi poprzez zbiór serwisów dostępnych dla narzędzi.



## ***Składnica informacji firmy Digital***

---

Składnica informacji oferuje rozproszone, aktywne serwisy dla aplikacji całego cyklu życia oprogramowania.

- ☐ **Tworzenie aplikacji**  
DECdesign, VAX Rdb/VMS, VAX DBMS, VAX C, VAX COBOL, VAX COBOL Generator, VAX RALLY, DECforms, itp.
- ☐ **Systemy działające - przetwarzanie transakcyjne**  
VAX ACMS, DECintact
- ☐ **Dostęp do informacji użytkowników końcowych**  
VAX DATATRIEVE, VAX TEAMDATA

## ***VAX CDD/Repository***

---

CDD/Repository jest pierwszą implementacją rozproszonej, aktywnej Składnicy informacji. CDD/Repository posiada otwarty interfejs umożliwiający tworzenie zintegrowanych aplikacji.

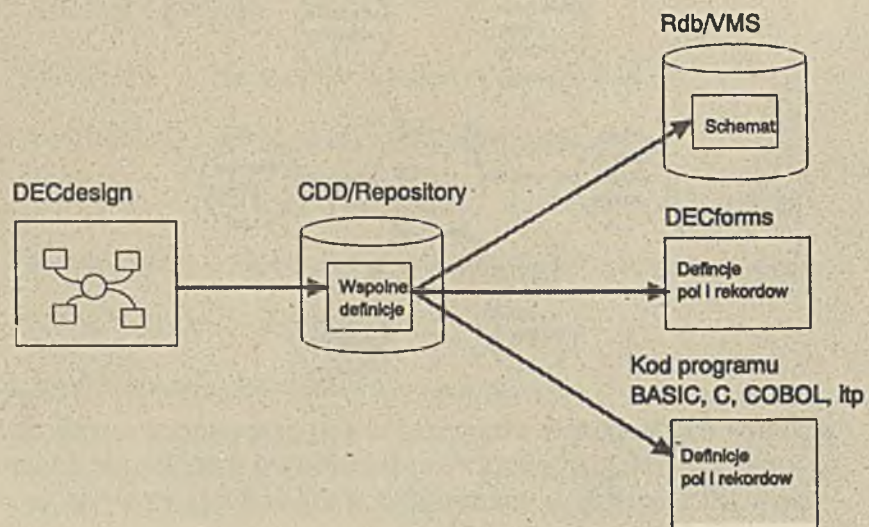
### **□ Charakterystyka CDD/Repository**

- Oparta o system zarządzania bazą danych (integralność, spójność, bezpieczeństwo danych)
- Związki pomiędzy definicjami (śledzenie i analiza wpływu zmian)
- Otwarty interfejs wywołań
- Rozproszenie w systemie VAXcluster i w sieci
- Interakcyjny interfejs do wprowadzania metainformacji i ich modyfikacji
- Zgodność z poprzednimi wersjami CDD
- Obiektowo-zorientowany interfejs oparty o ATIS
- Możliwość definiowania własnych reguł, obiektów i związków pomiędzy obiektami
- Dwie platformy (VMS i ULTRIX)



## *Integracja danych przez CDD/Repository gry*

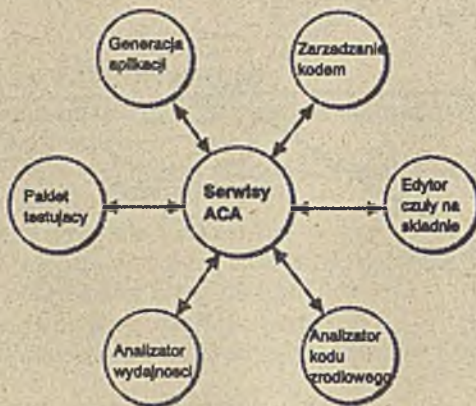
---



## *Współdziałanie aplikacji*

---

Serwisy ACA są kluczowym komponentem w dzieleniu zasobów pomiędzy aplikacjami działającymi w ramach architektury NAS. W szczególności architektura ta jest przydatna w środowisku narzędzi CASE.





## *Narzędzia środowiska COHESION*

---

### ☐ Analiza i projekt

DECdesign    Teamwork    Software Through Pictures

### ☐ Generatory kodu i języki czwartej generacji

VAX RALLY    VAX COBOL Generator    Power House    CorVision

### ☐ Zintegrowane środowiska wspomagające programowanie

DECset    Saber C    DECfuse

### ☐ Języki trzeciej generacji i kompilatory

Ada	APL	BASIC	BLISS32	C	COBOL
DIBOL	FORTRAN	LISP	OPS5	Pascal	
PL/I	RPG II	SCAN	DECTrellis		

## *Narzędzia środowiska COHESION c.d.*

---

### ☐ Cross-kompilatory

C      Ada      FORTRAN      Pascal

### ☐ Edytory i analizatory kodu

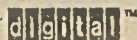
VAX LSE/SCA      xedit      vi      lint      ctrace

### ☐ Debuggery

VAX Debug      dxdb

### ☐ Zarządzanie kodem

VAX DEC/CMS      VAX DEC/MMS      sccs      make



Digital Equipment Polska



## *Narzędzia środowiska COHESION c.d.*

---

### ☐ **Narzędzia do testowania**

VAX DTM    VAX DEC/PCA    dxdiff    grep    pixi    DECtrace

### ☐ **Przygotowanie dokumentacji**

DECwrite    VAX DOCUMENT    Interleaf TPS    Uniplex

### ☐ **Przetwarzanie transakcyjne**

ACMS    DECintact

### ☐ **Zarządzanie informacją**

Rdb/VMS    VAX CDD/Repository    VAX Data Distributor  
VAX DBMS    VAX RdbAccess for ORACLE    VIDA for DB2  
VAX DATATRIVE    VAX TEAMDATA    DECdecision    Objectivity-DB



Digital Equipment Polska

COHESION, 29

## Narzędzia środowiska COHESION c.d.

Interfejs użytkowy				
Narzędzia cyklu życia				Zarządzanie
Planowanie	Analiza i projekt	Kodowanie	Utrzymanie	
Digital		DECdesign	VUIT	CDD/Administrator
		SSADM-SF	VAX RALLY	
		HOOD-SF	VAXtest / DECtest	
			DECPUSE	DECplan
			DECforms	DECwrite
			3gie	
	Inni producenci		PowerHouse	Softool
			Corvision	
		IEF	Focus	
		Teamwork	SmartStar	
STP		Gembase		
VSF				
CDD/Repository				
Network Application Support				



# CasePol

## ICASE - Zintegrowane środowisko CASE .

" Wyzwaniem dla współczesnej informatyki jest zamknięcie cyklu tworzenia aplikacji użytkowej od analizy aż do automatycznej generacji kodu źródłowego . "

James Martin

W czasie gdy w Polsce zmiany systemowe i strukturalne przebiegają w niezwykle dynamiczny sposób, informatyzacja przedsiębiorstw i urzędów administracji państwowej przysparza decydom przysłowiowego "ból głowy". Z jednej strony oferowane są gotowe rozwiązania, które nie zaspokajają w 100% zapotrzebowania systemowego przedsiębiorstwa. Z drugiej otwiera się ryzykowna, pełna pułapek droga tworzenia systemu "na miarę". Przewaga systemu "na miarę" jest niepodważalna. Zachodzi więc pytanie: jak zmniejszyć ryzyko i jak równocześnie przyspieszyć proces tworzenia aplikacji użytkowej. Tradycyjnym rozwiązaniem w tej sytuacji są języki czwartej generacji (4GL), systemy zarządzania bazą danych oraz strukturalne analiza i projektowanie. Pomimo niezaprzeczalnych zalet tych technik one też nie wypełniają w 100% zadania. Odpowiedzią na zadane wcześniej pytania jest w pełni zintegrowane środowisko CASE (Integrated Computer Aided Software Engineering - ICASE).

### Co to jest ICASE ?

Zintegrowane środowisko CASE wspomaga konstruktora (połączenie funkcji analityka i programisty) aplikacji użytkowych na każdym etapie cyklu tworzenia. Powszechnie przyjętym podziałem tego cyklu jest :

- Analiza
- Projektowanie
- Konstrukcja
- Wdrożenie
- Obsługa i Unowocześnienie.

Jeśli przyjrzeć się z bliska tym etapom oraz oferowanym na rynku narzędziom CASE, szybko zorientujemy się, iż dla prawie każdego etapu z łatwością możemy znaleźć po kilka mniej lub więcej udanych narzędzi. Trudności zaczynają się, gdy chcemy dobrać narzędzie działające we wszystkich lub w większości etapów. Wówczas okazuje się, że nawet te najbardziej znane nie spełniają założeń.

# CasePol

Dla pocieszenia mogę stwierdzić, iż na rynku światowym istnieje kilka narzędzi, które zbliżają się do tzw idealnego ICASE. Przykładami mogą być:

- CorVislon produkcji Cortex Corp.
- Change produkcji Change Inc.
- IEW produkcji Texas Instruments
- PowerCase produkcji Cognos.

Niektóre z nich wypełniają warunki strategii COHESION rekomendowanej przez DEC np CorVislon.

Rozpatrzmy teraz, w jaki sposób środowisko ICASE wspomaga poszczególne etapy cyklu tworzenia aplikacji użytkowej.

## Analiza.

W większości wypadków analizę możemy podzielić na tworzenie modelu bazy danych (bez względu czy używać będziemy systemu zarządzania bazą danych, czy nie) oraz określenie i sprecyzowanie poszczególnych funkcji występujących w aplikacji takich, jak : menu, formularze, wydruki, procedury etc. W obu przypadkach wiele środowisk ICASE oferuje graficzne interfejsy do graficznej reprezentacji modelu bazy danych jak również mapy funkcyjnej aplikacji. Dzięki takiemu przedstawieniu modelu bazy danych, np wg notacji J. Martina, środowisko ICASE umożliwia określenie relacji pomiędzy poszczególnymi zbiorami danych, które to w późniejszym stadium są użyte przy generacji kodu źródłowego odpowiedzialnego za operacje I/O. Mapa funkcyjna określa rozmieszczenie poszczególnych funkcji w hierarchii menu i ich typu np formularz, wydruk, procedura itp. Daje też możliwość określenia poziomu zabezpieczenia poszczególnych funkcji oraz określa nawigację wewnątrz aplikacji.

## Projektowanie/Konstrukcja

Gdy zostanie stworzona baza danych oraz mapa funkcyjna aplikacji, następuje równoczesny proces projektowania i konstrukcji poszczególnych funkcji. Odbyna się to przeważnie przy pomocy takich narzędzi jak screen painter (malowanie formularzy), raport painter (malowanie wydruków) oraz języków 4GL do pisania procedur. Narzędzia te usprawniają tworzenie interfejsów z użytkownikiem stosując filozofię "co wldzisz to otrzymasz", czyli wygląd projektowanych interfejsów zostaje przeniesiony do aplikacji bezpośrednio z narzędzia.



# CasePol

## Wdrożenie.

Po zakończeniu projektowania i konstrukcji aplikacji, lub jej części, następuje generacja kodu źródłowego. Jest to etap niezwykle istotny dla jakości powstającej aplikacji. Wysoko zaawansowane środowiska ICASE są w stanie wygenerować 95%-98% wysokiej jakości kodu źródłowego w 4GL (CorVision, PowerCase) lub w 3GL (język C - Change). Daje to pewność, że te 95%-98% jest wolne od błędów językowych i sprawdzać należy je tylko pod względem merytorycznym. Pozostałe "ręcznie" pisane 2%-5% podlega pełnemu zakresowi testów. Daje to efekt zmniejszenia tzw pola testu. Należy dodać, iż czołowe środowiska ICASE np CorVision generują tylko te moduły, które tego wymagają tzn takie, które zostały stworzone lub zmienione od czasu ostatniej generacji.

Kolejnym krokiem we wdrożeniu jest kompilacja kodu źródłowego. Niestety nie wszystkie środowiska ICASE pozwalają na ten krok. Wiele pozostaje przy typowym dla 4GL systemie interpretacyjnym, powodującym łatwo zauważalne zmniejszenie szybkości przetwarzania danych.

## Obsługa i Unowocześnienia.

Z badań przeprowadzonych w zachodnich kręgach informatycznych wynika, że 60%-70% kosztów poniesionych w czasie "życia" aplikacji, od chwili podjęcia tworzenia do chwili wycofania z eksploatacji, przypada na obsługę i unowocześnienia. Środowiska ICASE pozwalają na znaczną redukcję tych kosztów, dzięki temu, iż większość zmian dotyczy specyfikacji a nie kodu źródłowego. Ponieważ dane o aplikacji, tzw metadane, są zmagazynowane w składnicy (Repository), zmiana dowolnej definicji jest globalnie wprowadzona podczas następnej generacji. Zapewnia to ograniczenie potrzeby śledzenia i wprowadzania zmian w całej aplikacji, co w "ręczny" sposób może zająć miesiące.

Na zakończenie opisu cyklu tworzenia aplikacji w środowisku ICASE należy wspomnieć o możliwości automatycznej generacji dokumentacji aplikacji użytkowej.

# CasePol

## Metodologie.

Środowiska ICASE mogą współpracować z różnymi metodologiami. Jednakże najlepsze wyniki są uzyskiwane poprzez stosowanie metodologii opartej o założenia tzw RAD (Rapid Application Development). Przykładem takiej metodologii może być WIP (Wielokrotne Iteracyjne Prototypowanie). Metodologia ta zapewnia tworzenie wysokiej jakości aplikacji poprzez wspólną twórczą pracę przyszłych użytkowników i konstruktorów przy pełnym wykorzystaniu dynamizmu środowiska ICASE.

Na zakończenie mały przykład możliwości ICASE w połączeniu z WIP. Aplikacja użytkowa o następującej charakterystyce :

- 75 tabel (plików)
- 65 formularzy i menu
- 61 okienek
- 62 wydruki
- 5 procedur.

Ogólne oszacowanie aplikacji wyniosło 1800 punktów funkcyjnych. Aplikacja ta została wdrożona po 4 miesiącach, przy czym przez 1 miesiąc 1 analityk przeprowadził analizę wstępną, a następne 3 miesiące zespół 3 osobowy tworzył i wdrażał aplikację.

To jest możliwe !



mgr inż. Roman Łowkis  
Computer Systems  
for Business International Ltd  
02-119 Warszawa, ul. Pruszkowska 17  
40-955 Katowice, ul. Bytkowska 1b

# RELACYJNA BAZA DANYCH PROGRESS

## Ogólna charakterystyka systemu

### I. PROGRESS Software Corporation

#### 1. Ogólna charakterystyka firmy

PROGRESS Software Corporation jest amerykańską firmą zajmującą się produkcją oprogramowania z zakresu relacyjnych baz danych. Głównymi produktami oferowanymi przez PSC jest system zarządzania bazami danych (RDBMS) oraz język czwartej generacji PROGRESS 4GL. W skład oferty uzupełniającej wchodzi pakiet narzędziowy usprawniający proces tworzenia aplikacji, a nawet umożliwiające bezpośrednią obsługę bazy danych.

Oprócz oprogramowania PSC oferuje swoim klientom szczegółową dokumentację, a także rozbudowany system szkoleń i doradztwa technicznego. Jedną z podstawowych dewiz firmy jest duża dbałość o klienta. Ta cecha w powiązaniu z nowoczesnym stylem zarządzania, a przede wszystkim znakomitą pod względem technicznym produktem jest kluczem do sukcesu firmy. W katalogu 500 najszybciej rozwijających się amerykańskich firm prywatnych PROGRESS Software Corporation znalazła się w 1989 r. na 38 miejscu.

#### 2. Historia powstania i rozwoju PSC

PROGRESS Software Corporation powstała w grudniu 1981 roku. Jej głównym celem było stworzenie nowoczesnego narzędzia umożliwiającego obsługę relacyjnych baz danych.

Cel ten, którego końcowym efektem jest oferowany dziś PROGRESS Application Development System, realizowany był wieloetapowo. Równocześnie z rozwojem koncepcji merytorycznej systemu prowadzone były intensywne prace mające na celu maksymalne rozszerzenie bazy sprzętowej, na którą PROGRESS był oferowany.

Historię powstania dzisiejszej wersji PROGRESS-a wyznaczają następujące daty:

lipiec	'83	- początkowa wersja PROGRESS-a - etap testowania,
sierpień	'84	- pierwsza komercyjna wersja PROGRESS-a na UNIX,
styczeń	'85	- komercyjna wersja na MS-DOS,
czerwiec	'85	- wersje komercyjne na systemy XENIX, ULTRIX,
marzec	'86	- PROGRESS wersja 3,
kwiecień	'87	- sieciowa wersja PROGRESS-a na UNIX,

październik	'87	- PROGRESS na VAX/VMS, - PROGRESS wersja 4,
maj	'88	- PROGRESS pod CTOS/BTOS,
wrzesień	'88	- opublikowanie FAST TRACK-a,
maj	'89	- PROGRESS wersja 5,
lipiec	'90	- możliwość współpracy z bazami ORACLE oraz plikami RMS, - PROGRESS wersja 6,
listopad	'91	- PROGRESS for Windows, serwer NLM na Novella oraz mosty do narzędzi CASE (Excelerator i Knowledgware)
luty	'92	- bramy do AS/400, C-ISAM oraz CT-ISAM.

### 3. PSC i jej klienci

Ej klientem PSC to duża przyjemność. Od momentu nabycia pierwszego produktu i podpisania polisy serwisowej klient znajduje się pod ciągłą opieką firmy. W ramach polisy otrzymuje bezpłatnie pojawiające się regularnie nowe wersje nabytego produktu. Również bezpłatnie otrzymuje biuletyn techniczny PSC - wydawnictwo, w którym szczegółowo omawiane są wykryte błędy systemu, wyznaczony (i dotrzymywany) termin korekty, a także metody uniknięcia skutków. W biuletynie technicznym można też znaleźć informacje na temat wszystkich nowości w ofercie, planów rozwoju systemu, a także zmian organizacyjnych zachodzących w firmie.

Klient PSC ma prawo do bezpłatnego korzystania z telefonicznego serwisu technicznego czynnego w USA 24 godziny na dobę (w Polsce, z powodów organizacyjnych, tylko w godzinach pracy CSBI). Istnieje również możliwość bezpośredniego (on-line) połączenia się z PROGRESS Customer Bulletin Board - bazą danych zawierającą katalog technicznych problemów (wraz z rozwiązaniami), jakie kiedykolwiek zostały zgłoszone w dziale technicznym firmy.

Często wraz z rozwojem aplikacji, rozrostem bazy danych lub liczby użytkowników zachodzi konieczność wymiany sprzętu. Z tym również nie ma problemu. Klient nie napotyka praktycznie żadnych ograniczeń w wyborze sprzętu (patrz p.III.2). Koszty takiej operacji równają się jedynie różnicy cen pomiędzy starą, a nową wersją produktu.

Poza realizacją zobowiązań wynikających z polisy serwisowej, PSC prowadzi szeroką działalność w zakresie kontaktów z klientami oraz pomiędzy klientami. Takim celom służy sieć klubów użytkowników PROGRESS-a (PROGRESS Users Groups) obejmująca USA, Australię oraz dziewięć krajów europejskich w tym (od maja 1992) Polskę. Co roku użytkownicy mają okazję spotkać się na specjalnie organizowanych meetingach, gdzie oprócz bezpośredniego kontaktu z twórcami PROGRESS-a mają okazję zaprezentowania własnych aplikacji i projektów związanych z PROGRESS-em. Corocznie wydawany jest katalog aplikacji progressowych, zawierający ponad 1000 pozycji. Każdy użytkownik PROGRESS-a może zamieścić w nim darmowy opis swojego produktu, co - biorąc pod uwagę ogólnosiątkowy zasięg wydawnictwa i jego wielotysięczny nakład - jest wspaniałą metodą promocji swoich aplikacji.

Dużo informacji na temat PSC, a także powstających firm satelickich, uzyskać można prenumerując "Profiles" - ilustrowany magazyn PSC wydawany w cyklu miesięcznym.

### 4. PSC w oczach innych

Od początku swej kariery rynkowej PROGRESS Software Corporation stanowi obiekt wnikliwego zainteresowania zarówno konkurencji, jak i niezależnych organizacji, które zajmują się oceną światowego rynku oprogramowania. Do takich organizacji należy DATAPRO, która w swych raportach z lat 1988, 1989, 1990, 1991 umieszcza PROGRESS na pierwszym miejscu w swojej klasie, pośród takich systemów jak SYBASE, ORACLE, INGRESS, INFORMIX, czy FOCUS. Należy zaznaczyć, że DATAPRO ocenia nie tylko produkt, ale również całokształt usług oferowanych przez firmę. Na ocenę ogólną mają więc wpływ takie czynniki jak jakość dokumentacji firmowej, zakres prowadzonych szkoleń, zakres udzielanej licencji serwisowej, a także poziom i dyspozycyjność serwisu technicznego. Zwycięstwo w tej klasyfikacji jest prawdziwym powodem do dumy, choć nie stanowi zaskoczenia dla osób, które zetknęły się z PSC jako pracownicy lub klienci.



## 5. PSC w Polsce

Jeszcze do niedawna nie można było mówić o żadnej obecności PROGRESS-a w Polsce, ponieważ na produkty tak zaawansowanej technologii, jaką reprezentuje ten system obowiązywało embargo COCOM-u. Począwszy od ubiegłego roku, kiedy embargo to zostało zniesione, nastąpiła gwałtowna ekspansja PROGRESS-a na rynki wschodnioeuropejskie. PROGRESS wkroczył już do Polski, Czechosłowacji i Węgier, a w najbliższej przyszłości należy oczekiwać jego wejścia na pozostałe rynki wschodniej Europy, nie wyłączając rynku radzieckiego.

W Polsce interesy PROGRESS Software Corporation reprezentuje CSBI (Computer Systems for Business International) - polsko-brytyjskie przedsiębiorstwo joint venture będące równocześnie dystrybutorem sprzętu, a także gotowych aplikacji napisanych w PROGRESS-ie. Oprócz funkcji handlowych firma CSBI zapewnia użytkownikom PROGRESS-a doradztwo techniczne, organizuje seminaria i pokazy z udziałem przedstawicieli PSC. Z drugiej strony dzięki naciskom (a także dużemu nakładowi pracy) ze strony CSBI powstała już częściowo spolonizowana wersja PROGRESS-a, a w najbliższym czasie ukaże się pełna polska wersja narodowa PROGRESS-a.

Polska lista referencyjna zawiera już kilkadziesiąt pozycji, obejmujących banki, wyższe uczelnie, zakłady przemysłowe, firmy informatyczne a także urzędy państwowe.

## II. Produkty oferowane przez PSC

### 1. PROGRESS 4GL & RDBMS

System zarządzania relacyjną bazą danych i język czwartej generacji PROGRESS są flagowymi produktami PROGRESS Software Corporation. Stanowią one komplet narzędzi potrzebnych do utworzenia i uruchomienia złożonej, wieloużytkownikowej aplikacji pracującej praktycznie na dowolnej platformie sprzętowej.

Produkty te dostarczane są w zintegrowanym pakiecie obejmującym również interakcyjny słownik danych, edytor z analizatorem składni, system bieżących odpowiedzi oraz bibliotekę gotowych procedur. Na szczególną uwagę zasługuje słownik danych, który oprócz swych podstawowych funkcji - definicji struktury bazy i indeksów - umożliwia interakcyjne operacje na danych, takie jak: DUMP/LOAD, import i eksport plików w zadeklarowanym formacie, import definicji struktury oraz danych bezpośrednio z plików DBASE, przyłączenie lub odłączenie dowolnej bazy, a także zamrożenie części danych. Słownik umożliwia też utworzenie struktury użytkowników i administratorów bazy, a także generację różnorodnych raportów.

### 2. PROGRESS FAST TRACK

Jest rozbudowanym generatorem aplikacji stanowiącym pierwszy krok w technologii CASE w odniesieniu do PROGRESS-a. Jego cztery główne składowe, to edytor menu, edytor formatek ekranowych, projektant raportów i generator procedur QBF (Query By Form).

Menu Editor (edytor menu) umożliwia błyskawiczne projektowanie dowolnych struktur menu, a także przypisywanie poszczególnym punktom różnych funkcji, od standardowych (return, quit, shell), aż do uprzednio stworzonych procedur. W ten sposób możliwe jest używanie funkcji FAST TRACK-a jednocześnie z "klasycznie" utworzonymi fragmentami aplikacji.

Screen Painter (edytor formatek) umożliwia komponowanie ekranu poprzez rozmieszczanie na nim pól bazy, zmiennych wraz z etykietami i komentarzami. Dzięki filozofii WYSIWYG (What-You-See-Is-What-You-Get)<sup>1</sup> narzędzie to jest wygodne nawet dla początkujących użytkowników.

---

<sup>1</sup> ang. dosł. co widzisz, to masz.

Zaprojektowana formatka może zostać wykorzystana do wygenerowania procedury QBF lub zapisana na dysku w postaci pliku ASCII do wykorzystania w dowolnej procedurze jako plik włączony.

QBF - "zapytanie poprzez formatkę" jest pojęciem wprowadzonym przez twórców PROGRESS-a. Procedura QBF (zawsze związana z konkretną formatką) umożliwia przegląd bazy (formatka po formacie), wyszukiwanie żądanej informacji poprzez konstruowanie dowolnie złożonych warunków logicznych (system dba o ich poprawność), wprowadzanie, kasowanie i modyfikację informacji, generację raportu, a także przejście do innej procedury QBF (powiązanej z poprzednią np. poprzez wartość pola).

Report Writer (projektant raportów) umożliwia, w podobny sposób jak edytor formatki, zaprojektowanie szaty graficznej i zawartości raportów, które będą generowane w systemie.

Całość systemu przechowuje zadeklarowane struktury (formatki, menu i raporty) w specjalnie do tego celu rozszerzonej strukturze bazy danych. Dzięki temu w każdej chwili możliwe jest powrót do pracy nad projektem, uruchomienie dowolnego jego fragmentu, a także wygenerowanie tekstu odpowiedniej procedury w 4GL. W rezultacie uzyskujemy bardzo silne narzędzie do tworzenia kompletnych aplikacji, szczególnie zalecane do tworzenia prototypów funkcjonalnych.

### 3. PROGRESS RESULTS

PROGRESS RESULTS jest jednym z najnowszych produktów PSC. Jego główną ideą jest umożliwienie obsługi każdej bazy danych bez konieczności pisania kosztownej, dedykowanej aplikacji. Absolutna uniwersalizacja funkcji takiego pakietu jest oczywiście niemożliwa. PROGRESS RESULTS stanowi tu rozsądny kompromis.

Składa się z pięciu podstawowych modułów:

- |                |   |
|----------------|---|
| Query          | - umożliwia przegląd i modyfikację informacji na poziomie rekordu, wykorzystuje technikę zapytań QBE (Query By Example);                                |
| Reports        | - pozwala ujmować zawarte w bazie informacje w dowolne zestawienia;   |
| Labels         | - automatyczny wydruk etykiet (adresy, identyfikatory, etykiety produktu);  |
| Data Export    | - segment transferu informacji do innych pakietów software'owych (w formatach ASCII, DIF, SYLK, Wordstar, MS-Word, WordPerfect, LOTUS oraz OfisWriter); |
| Administration | - pozwala dostosować RESULTS do środowiska i potrzeb konkretnej aplikacji.  |

Moduły te obsługują większość typowych potrzeb użytkowników systemów bazodanowych. Wszystkie nietypowe zapytania mogą być zaspokojone poprzez połączenie RESULTS z dedykowaną aplikacją użytkownika. Struktura pakietu umożliwia zarówno włączenie niezależnych procedur do menu RESULTS, jak i wywołanie RESULTS z menu aplikacji.

### 4. Pozostałe pakiety

PROGRESS RUN-TIME jest minimalnym środowiskiem niezbędnym do korzystania z gotowej aplikacji progressowej. Umożliwia on uruchomienie aplikacji w postaci wynikowej, bez dostarczania użytkownikowi kodu źródłowego. Nie daje natomiast możliwości pisania własnych aplikacji ani też modyfikacji struktury bazy. Jego cena jest kilkakrotnie niższa od ceny PROGRESS 4GL & RDBMS.

PROGRESS DEVELOPER'S TOOLKIT jest pakietem narzędziowym umożliwiającym programiście przeprowadzenie adjustacji gotowej aplikacji u użytkownika. Możliwości TOOLKIT-a obejmują:



- rekompilację aplikacji w nowym środowisku (bez modyfikacji kodu!),
- zmodyfikowanie metaschematu bazy,
- nałożenie ograniczeń na dostęp do definicji struktur bazy.

Pakiet ten jest potrzebny jedynie programistom dostarczającym oprogramowanie w PROGRESS-ie. Użytkownik aplikacji lub programista piszący dla własnych potrzeb nie muszą zaopatrywać się w to narzędzie.

PROGRESS TEST-DRIVE jest w pełni funkcjonalną wersją PADS (patrz p. 5) zawierającą pewne wbudowane ograniczenia ilościowe (liczba dostępów do tej samej bazy). Jest ona rozprowadzana po symbolicznych cenach w celach reklamowo - szkoleniowych. Zakupienie TEST DRIVE-a jest najlepszą metodą na poznanie PROGRESS-a i znalezienie odpowiedzi na pytanie, czy produkt ten odpowiada naszym potrzebom. Wbudowane ograniczenia nie pozwalają jednak wykorzystać go jako docelowego narzędzia obsługi bazy danych.

## 5. Zestawy pakietów

Pakiety oferowane przez PSC uzupełniają się nawzajem, dlatego też oferowane są w zestawach, których cena łączna jest niższa od sumy cen poszczególnych składników. Podstawowym takim zestawem jest PROGRESS APPLICATION DEVELOPMENT SYSTEM (PADS). Składa się on z PROGRESS 4GL & RDBMS oraz FAST TRACK-a.

Drugi zestaw to PROGRESS QUERY/REPORT. Nie jest on jednak prostą sumą paru innych produktów. Oparty o PROGRESS RUN-TIME nie daje możliwości modyfikacji struktury bazy ani też pisanie w 4GL. Udośćpnia natomiast pełne możliwości PROGRESS RESULTS oraz modułu Report Writer pakietu FAST TRACK. Zestaw ten stanowi minimum niezbędne do utr. mania własnej bazy danych przy maksymalnej redukcji kosztów.

## III. System PROGRESS - charakterystyka techniczna

### 1. Wstęp

PROGRESS 4GL & RDBMS jest nowoczesnym, szybkim i bezpiecznym narzędziem służącym do zarządzania dużą ilością danych w rozproszonym i niejednorodnym środowisku systemowym. Podstawowa jego cecha to niemal całkowita niezależność od sprzętu, systemu operacyjnego czy konfiguracji sieciowej, w której pracuje. Jest to możliwe dzięki opracowaniu wersji systemu na kilkaset platform sprzętowych, obejmujących wszystkich czołowych producentów oraz różne opcje systemowe, a także dzięki implementacji najczęściej spotykanych protokołów sieciowych.

Język PROGRESS 4GL należący do grupy języków czwartej generacji jest bardzo wydajnym i elastycznym narzędziem pozwalającym skrócić czas pisania aplikacji około dziesięciokrotnie w stosunku do języków trzeciego poziomu. Zapewniając pełną kontrolę klawiatury oraz ekranu umożliwia on dostosowanie interfejsu użytkownika do indywidualnych potrzeb i przyzwyczajeń. Zastosowane mechanizmy wspomagania programisty zwiększają komfort pracy a elastyczna struktura systemu ułatwia wprowadzanie zmian już po zamknięciu projektu.

System jako całość wyposażony jest w rozbudowany mechanizm ochrony danych. Obejmuje on ochronę przed niepożądanym dostępem, kontrolę spójności logicznej informacji oraz wielopoziomowe zabezpieczenia danych przed fizycznym uszkodzeniem.

Wydajność systemu w środowisku wielodostępnym zapewniona jest przez zastosowanie najnowocześniejszych technik przetwarzania oraz maksymalne wykorzystanie możliwości sprzętowych (n.p. w przypadku wieloprocesorowych maszyn Sequenta).

## 2. Środowisko pracy PROGRESS-a

### a) Platformy sprzętowe

Niezależność od sprzętu jest jedną z najważniejszych zalet PROGRESS-a. Oznacza ona pełną przenośność aplikacji pomiędzy różnymi platformami sprzętowymi, a także możliwość łączenia różnych systemów w spójne środowisko, w którym przetwarzanie rozproszone, a także dostęp do nielokalnych (również wielowolumenowych) baz danych kontrolowany jest przez PROGRESS.

Lista dostępnych dla PROGRESS-a platform sprzętowych jest prawdopodobnie najszerza spośród list systemów konkurencyjnych. W chwili obecnej obejmuje ona ponad 200 (a z uwzględnieniem różnych wersji maszyn - ponad 500) pozycji. Obecne są na niej produkty takich firm jak: ALR, Altos, Apple, Apricot, Arix, AT&T, Bull, Compaq, Data General, DDE, DEC, Dell, DIAB, Dolphin, Edisa, Everex, Harris, Hewlett-Packard, IBM, ICL, Intel, ITL, MIPS, Motorola, NCR, Nixdorf, Nokia Data, Norsk Data, Olivetti, Opus Systems, Philips, Prime, Pyramid, Sanyo/Icon, Sequent, SID INFORMATICA, Siemens, Silicon Graphics, Solbourne, Sun, Tandem Integrity, Texas Instruments, TIS Limited, UNISYS, ZENITH.

Różnorodne architektury komputerów znajdujących się na tej liście oparte są o procesory M680X0, 80X86, RISC, VAX, 88000 oraz SPARC.

### b) Systemy operacyjne

PROGRESS pracuje pod kontrolą takich systemów operacyjnych jak DOS, OS/2, UNIX i jego pochodne (XENIX, ULTRIX, AIX, A/UX, HP-UX i.t.d.), VMS, BTOS, CTOS.

Do uruchomienia PROGRESS-a w środowisku UNIX wymagane jest min. 2 MB pamięci operacyjnej. Minimalne wymagania co do środowiska DOS-owego są następujące:

- system w wersji 3.1 lub wyższej,
- dysk twardy (min. 17MB miejsca dla PADS + miejsce na bazę i aplikacje),
- stacja dysków miękkich dużej gęstości (DS/HD),
- 640 KB pamięci operacyjnej (wersja pamięci oszczędna),

2 MB pamięci operacyjnej (wersja pełna).

Istnienie wersji pamięci oszczędnej na DOS (całkowicie równoważnej funkcjonalnie w stosunku do wersji pełnej) jest cechą wyróżniającą PROGRESS spośród innych systemów tej klasy. Jest to niesłychanie istotne z punktu widzenia użytkowników, którzy chcieliby zainstalować system na już posiadanym sprzęcie, bez konieczności jego modyfikacji. Oczywiście użytkowanie tej wersji jest niewygodne, pozwala jednak rozłożyć w czasie koszty związane z zakupem nowego systemu. Wersje pełna i pamięci-oszczędna sprzedawane są jednocześnie, a więc w przypadku dokupienia pamięci cała operacja zmiany wersji sprowadza się do przestawienia jednej zmiennej systemowej.

### c) Praca w sieciach

Mocną stroną PROGRESS-a są jego możliwości pracy w środowisku sieciowym. Możliwości te obejmują pracę w sieciach jednorodnych opartych na protokołach TCP/IP, NetBios, SPX/IPX oraz DECnet, a także pracę w sieciach heterogenicznych.

Praca w DOS-owych sieciach rozległych sprowadza się do komunikacji procesu klienta z odległym serwerem bazy danych za pośrednictwem modułu CCM<sup>2</sup>. Możliwe jest przy tym wykorzystanie wszystkich wymienionych protokołów równocześnie. W sieciach lokalnych PROGRESS umożliwia dostęp do bazy ze wszystkich węzłów sieci, a także obsługę żądań odległych poprzez CCM z wykorzystaniem protokołów NetBios oraz SPX/IPX. Proces serwera, a także procesy klientów, mogą zostać uruchomione na dowolnym komputerze w sieci, jednakże - ze względu na jednozadaniowość DOS-a - nie jest możliwe uruchomienie wielu procesów na jednej maszynie.

---

<sup>2</sup> skr. ang. Client Communication Module - moduł komunikacyjny klienta.



Najnowszym produktem PROGRESS-a (wersja 6.2H) jest moduł NLM<sup>3</sup>. Jego zastosowanie w sieciach Novell (wersje od 3.11) daje znaczne korzyści czasowe dzięki lepszemu wykorzystaniu możliwości sprzętowych.

PROGRESS pracuje z dwoma wersjami protokołu TCP/IP - firm Excelan oraz FTP Software. Interfejs Excelan obsługiwany jest przez wbudowane procedury PROGRESS-a. Aby wykorzystać wersję FTP, konieczne jest zainstalowanie właściwego programu obsługi.

W celu przyłączenia PC do bazy danych pracującej pod VMS należy zakupić DECnet DOS Digital Equipment Corporation. Oprogramowanie to tłumacza wywołania NetBios-a na wywołania protokołu DECnet, co umożliwiła serwerowi VMS obsługę żądań klienta DOS-owego.

Większość instalacji systemu UNIX posiada wbudowany protokół TCP/IP. Jest on wykorzystywany przez PROGRESS. Dla pozostałych wersji UNIX-a i XENIX-a PROGRESS zakłada wykorzystanie TCP/IP z interfejsem Excelan. Karty sieciowe i oprogramowanie należy zakupić wówczas oddzielnie w firmie Excelan. Najnowszym produktem PROGRESS-a jest moduł umożliwiający łączenie instalacji UNIX-owych z siecią Novell.

Implementacja protokołu sieciowego DECnet pozwala na tworzenie sieci pod VAX/VMS. Poprzez DECnet DOS możliwe jest przyłączenie DOS-owych klientów do serwera bazy pracującego pod VAX/VMS. PROGRESS daje również możliwość rozdziału aplikacji pomiędzy klastery VAX-a.

Instalacje PROGRESS-a w środowisku CTOS/BTOS, ze względu na cechy wbudowane tych systemów, nie wymagają żadnych sieciowych interfejsów (wersja sieciowa jest wersją standardową). Komunikacja zewnętrzna odbywa się za pośrednictwem protokołu TCP/IP. Wewnątrz sieci BTOS/CTOS możliwe jest również wykorzystanie BNET, CTNET oraz APNET.

Serwer PROGRESS-a pracujący pod kontrolą OS/2 obsługuje zgłoszenia klientów sieci za pośrednictwem protokołów NetBios lub SPX. Ponieważ klienci DOS-owi mogą pracować z wykorzystaniem tych samych protokołów, mają oni wygodny dostęp do informacji zgromadzonej w bazach pozostających pod kontrolą OS/2.

#### d) Współpraca z systemami okien

Ekspansja systemów okienkowych powoduje coraz powszechniejsze dostosowywanie oprogramowania do ich wymogów. PROGRESS posiada wbudowane możliwości współpracy z GUI<sup>4</sup> bazującymi na standardzie X-Windows. Należą do nich MOTIF, Open Look, Open Desktop, DEC Windows oraz Presentation Manager. Oprócz tego PROGRESS potrafi współpracować z NewWave Hewlett-Packarda, a ostatnio ujrzała światło dzienne wersja przeznaczona dla Microsoft Windows 3.0.

Istotną cechą systemu PROGRESS jest to, że istniejące aplikacje nie pisane z myślą o systemach okienkowych mogą bez żadnych zmian zostać na nich uruchomione. Wbudowane cechy PROGRESS-a zapewnią bezbłędną obsługę wszystkich funkcji takich systemów. Należy jedynie zauważyć, że zastosowanie systemu MS-Windows 3.0 podnosi wymagania sprzętowe instalacji. Komputer z taką konfiguracją oprogramowania wymaga już wersji DOS-a 3.3 lub wyższej, minimum procesora 80286, chociaż zalecany jest 80386 i wyższe oraz min. 2 MB pamięci operacyjnej (zalecane 4 MB).

#### e) Import i eksport danych

System PROGRESS wyposażony jest w wygodny mechanizm wymiany danych z otoczeniem. Z poziomu aplikacji możliwy jest import oraz eksport danych z/do plików ASCII o określonym formacie. Jeżeli plik nie spełnia wymogów formatu PROGRESS-a można go przetworzyć za pomocą programu QUOTER. W trakcie pracy interaktywnej (jedna z opcji Data Dictionary) możliwy jest eksport danych w formacie ASCII, DIF (Data Interchange Format), SYLK

<sup>3</sup> skr. ang. Network Loadable Module - moduł rezydujący jako zadanie dodatkowe na serwerze sieci Novell.

<sup>4</sup> skr. ang. Graphical User Interface - graficzny interfejs użytkownika.

(SYmbolic LiNK)<sup>6</sup>, WordPerfect, WordStar, Word oraz BTOS OfisWriter. Możliwy jest też import danych w formacie DIF, SYLK, ASCII<sup>6</sup>. W PROGRESS Data Dictionary istnieje również opcja pozwalająca na import definicji struktury oraz danych bezpośrednio z bazy DBASE (II, III, III+, IV).

Inną metodą komunikacji PROGRESS-a z otoczeniem są potoki<sup>7</sup>. Są one jednym z mechanizmów komunikacji międzyprocesowej dostępnym w systemie operacyjnym UNIX. Stwarzają duże możliwości przekazywania informacji pomiędzy aplikacją PROGRESS-ową, a programami napisanymi w innych językach. Poprzez potok UNIX-owy możliwe jest na przykład wysłanie zapytania SQL do aplikacji PROGRESS-owej i uzyskanie w odpowiedzi żadanego zestawu danych z bazy PROGRESS-a.

Wykorzystanie potoków stanowi alternatywę w stosunku do niektórych zastosowań interfejsu HLC umożliwiającego wywołanie PROGRESS-a z poziomu języka trzeciej generacji.

### 3. Baza danych

#### a) Organizacja bazy PROGRESS

Omawiając strukturę bazy danych w systemie PROGRESS, należy zacząć od definicji podstawowych pojęć. Nie zawsze bowiem nazwa danego elementu będzie miała znaczenie zgodne z naszymi przyzwyczajeniami. Mankament ten dotyczy niemal wszystkich istniejących systemów, stąd najlepsze wydaje się pozostanie w obrębie terminologii ustalonej przez twórców systemu, którego używamy.

Najmniejszą jednostką informacji jest w PROGRESS-ie pole rekordu. Jest ono ekwiwalentem zmiennej języka i może przyjmować te same co zmienna typy (Character, Integer, Logical, Date, Raw oraz RecID). Z założenia tego wynika nietypowa dla innych systemów możliwość przyjmowania przez pole wartości tablicowej. Pole jest wówczas wektorem zmiennych (w takim sensie jak n.p. tablica w PASCAL-u).

Zestaw pól tworzy rekord. Inne nazwy rekordu to wiersz relacji lub krotka. Bardzo ważną cechą PROGRESS-a jest zmienna długość rekordu (a więc i pola). Deklarowany format pola jest tylko formatem stosowanym do operacji wejścia/wyjścia. Informacja pamiętana jest w bazie w całości, nawet jeżeli długość zapisu przekracza aktualnie zadeklarowaną.

Zbiór rekordów nazywany jest w PROGRESS-ie plikiem (ang. file), co może prowadzić do nieporozumień, ponieważ nie ma on nic wspólnego z plikiem systemowym. Plik PROGRESS-a należy rozumieć jako tablicę (termin SQL) lub relację.

Baza danych jest z kolei zbiorem plików. Całość bazy mieści się w jednym pliku systemowym. W pliku tym - oprócz samych danych - pamiętane są definicje struktury bazy, indeksów, etykiety i opisy pól rekordów, warunki dostępu, a nawet klauzule walidacji sprawdzane w trakcie modyfikacji pól bazy. Takie rozwiązanie ogranicza liczbę odwołań do systemu operacyjnego, a więc jest korzystne ze względu na efektywność i niezawodność systemu.

#### b) Słownik danych

Bardzo istotnym elementem systemu PROGRESS jest słownik danych (Data Dictionary). Słownik ten umożliwia definicję i modyfikację struktury bazy, przegląd zdefiniowanych struktur SQL, przyłączenie i odłączenie bazy, wybór bazy aktywnej, generację plików różnicowych, generację

---

<sup>6</sup> co umożliwia wymianę danych z pakietami takimi jak: Lotus 1-2-3, Excel, Uniplex, MultiPlan, 20/20.

<sup>7</sup> możliwe są dwie opcje:

- z wyróżnionymi ogranicznikami pól i rekordów,
- ze sztywnym formatem.

<sup>7</sup> ang. Named Pipes, inaczej: kolejki FIFO - First In First Out.



plików błędów, import i eksport danych i struktur, wprowadzenie struktury użytkowników i ograniczeń dostępu, a także generację różnorodnych raportów. Wszystkie funkcje słownika dostępne są w trybie interaktywnym dzięki wygodnemu Pull-Down Menu. Są one pogrupowane w następujące podmenu:

### **Modify-Schema**

Znajdują się tu funkcje umożliwiające definiowanie i modyfikowanie schematu bazy danych. Definiuje się strukturę plików w bazie, strukturę rekordu w danym pliku oraz indeksy.

Możliwa jest zmiana struktury istniejącej bazy bez naruszenia zawartych w niej danych, jednakże nie można zmienić typu zadeklarowanego pola. Tego rodzaju operację należy przeprowadzić etapowo, definiując nowe pole, przepisując dane ze starego pola za pomocą odpowiedniej procedury konwersji, po czym usuwając stare pole. Ze względu na zmienną długość rekordu w bazie PROGRESS-a wszelkie zmiany formatu pola są dozwolone i nie mają wpływu na informację przechowywaną w bazie.

Indeks może zostać utworzony poprzez konkatenację pól. Może on być zadeklarowany jako unikalny. System nie dopuszcza wówczas do powstania dwóch rekordów o tej samej wartości klucza, może też być zadeklarowany jako skrócony, co umożliwi wyszukiwanie rekordów o kluczu zaczynającym się od podanego ciągu znaków. Każde pole wchodzące w skład klucza może mieć przypisaną prostą lub odwróconą regułę kolejności.

### **SQL**

Podmenu SQL zawiera funkcje umożliwiające przegląd zdefiniowanych struktur SQL. Można wyświetlić listę istniejących projekcji<sup>8</sup>, obejrzeć ich struktury, a także wygenerować fragment procedury napisanej w SQL DDL<sup>9</sup> odpowiadającej danej projekcji.

### **Database**

Funkcje zawarte w tym podmenu pozwalają na przyłączenie i odłączenie bazy danych, wybór bazy aktywnej, utworzenie nowej, pustej bazy danych oraz na wykonanie niektórych szczególnych operacji specyficznych dla poszczególnych typów baz (PROGRESS, AS/400, ORACLE, Rdb, RMS). Operacje na bazach nie progressowych stają się jednak dostępne dopiero po zakupieniu dodatkowych modułów (bram).

Dla baz progressowych możliwe jest utworzenie pliku różnicowego pomiędzy bazami (plik ten może posłużyć później do uzupełnienia brakujących danych w bazie, lub też do innych celów), konwersja pliku różnicowego z wersji 5 na 6, utworzenie pliku z danymi w formacie wersji 5 oraz utworzenie listy parametrów domyślnych baz automatycznie przyłączanych przez system w trakcie wykonywania aplikacji.

### **Admin**

Podmenu Admin[istration] obsługuje wszystkie funkcje związane z importem i eksportem danych oraz systemem bezpieczeństwa<sup>10</sup>. Tutaj również umieszczone zostały funkcje zrzutu i odtworzenia stanu bazy (DUMP/LOAD Data Def. & File Contents). W trakcie odtwarzania stanu bazy możliwa jest dezaktywacja indeksów (względny czasowy) oraz utworzenie specjalnego pliku rekordów błędnych, który, po poprawieniu za pomocą dowolnego edytora, można później wprowadzić jako plik różnicowy.

### **Utilities**

Podmenu narzędziowe zawiera edytor zbiorów parametrów (\*.pf), interakcyjny interfejs do programu Quoter - formatera plików, generator plików włączonych obsługujący najczęściej spotykane zastosowania instrukcji ASSIGN, FORM oraz DEF WORKFILE języka 4GL. Tu umieszczona jest funkcja Info dostarczająca skondensowanej informacji o stanie systemu, jak również funkcja Freeze/Unfreeze pozwalająca na tymczasowe zamrożenie dowolnego pliku bazy

---

<sup>8</sup> ang. views.

<sup>9</sup> skr. ang. Data Definition Language - język definicji danych.

<sup>10</sup> patrz rozdz. III.2.e oraz III.6.a.

danych.

### Reports

Tu zgromadzone są funkcje umożliwiające przegląd i wydruk (możliwe jest zapisanie w pliku) informacji ogólnych na temat użytkowanej bazy. Można więc uzyskać szczegółowy raport na temat zdefiniowanych plików i struktur rekordów tych plików, listę samych plików, listę pól rekordu w danym pliku, strukturę indeksów, listę użytkowników, listę zdefiniowanych projekcji oraz listę implikowanych (poprzez nazwę) relacji.

#### c) Bazy innych systemów

PROGRESS 4GL jest językiem czwartej generacji niezależnym od bazy danych. Może on zostać wykorzystany do obsługi baz danych innych systemów. W chwili obecnej istnieją bramy umożliwiające bezpośrednią obsługę z poziomu 4GL struktur ORACLE, RDB, RMS, AS/400 oraz C-ISAM. Struktury te (w postaci baz lub plików nierelacyjnych) rozmieszczone na rozproszonym i niejednorodnym sprzęcie, znajdujące się pod kontrolą różnych systemów operacyjnych i połączone za pomocą różnych protokołów sieciowych mogą być wykorzystywane jednocześnie, przez jedną aplikację, a dane z nich pochodzące mogą być modyfikowane w jednej formacie ekranowej. Całość działań przebiega pod kontrolą 4GL, a więc nie zachodzi potrzeba odwoływania się do języków niższego poziomu. Prowadzi to do uzyskania stopnia spójności aplikacji niedostępnego w innych systemach.

#### d) Ograniczenia liczbowe

Dewiza PROGRESS Software Corporation jest "Usual limits don't apply"<sup>11</sup>. Najlepszą weryfikacją tej tezy jest przegląd ograniczeń liczbowych systemu PROGRESS w wersji 6:

Rozmiar bazy:	200 GB,
Liczba aktywnych baz:	240 baz jednocześnie,
Długość rekordu:	32000 B,
Długość pola:	32000 B,
Liczba pól w rekordzie:	Jedyne ograniczenie wynika z długości rekordu,
Liczba plików w bazie:	1023,
Liczba indeksów / plik:	1023,
Liczba indeksów w bazie:	1023,
Całkowita liczba użytkowników:	Zależnie od sprzętu, maksymalnie 2048,
Typy danych:	
character	maks. 32000 B, pełny zbiór ASCII,
decimal	maks. 50 cyfr, w tym do 10 miejsc po przecinku,
integer	od -2 147 483 648 do +2 147 483 647,
logical	true/false, yes/no lub w/g deklaracji,
date	1.1.32768 p.n.e. do 12.31.32767 n.e.
null	istnieje możliwość posługiwania się wartością pustą.

Zamieszczone zestawienie jest dużym skrótem szczegółowego wykazu ograniczeń dostępnego w Progress Help (p. IV.3).

---

<sup>11</sup> ang. zwykle ograniczenia dla nas nie istnieją.



## 4. Język

### a) PROGRESS 4GL

PROGRESS 4GL jest językiem czwartej generacji pomyślanym jako uniwersalna nadbudowa nad mechanizmem zarządzania bazą danych (RDBMS). Elastyczność ta została osiągnięta poprzez przemysłowy dobór poleceń języka oraz zastosowanie mechanizmu bram pośredniczących w transferze danych z i do bazy. Użytkownik korzystający z PROGRESS 4GL może więc obsługiwać wiele różnych baz jednocześnie w jednej spójnej aplikacji.

Określenie język czwartej generacji w odniesieniu do PROGRESS 4GL oznacza, że jest to język raczej deklaracyjny niż sekwencyjny. Innymi słowy użytkownik deklaruje w nim chęć osiągnięcia określonego celu zamiast kodowania sekwencji operacji do niego prowadzących. Metody osiągania tego celu są już domeną systemu. Na obecnym etapie rozwoju języków komputerowych cechy deklaratywne nie eliminują sekwencyjnego układu programu. Jest on zresztą zgodny z naturalnym sposobem analizy problemu przez człowieka.

Składnia 4GL jest wyjątkowo przejrzysta i konsekwentna. Mimo iż w wielu przypadkach liczba parametrów pojedynczego polecenia może być bardzo duża, w praktyce większość z nich jest przyjmowana bądź wyliczana automatycznie na podstawie bieżących parametrów systemu. Posługiwanie się wartościami domyślnymi jest kluczem do siły tego języka. Umożliwia ono programiście skoncentrowanie się na schemacie funkcjonalnym projektowanego systemu bez tracenia czasu na szczegóły. Pielęgnacja strony użytkowej aplikacji może nastąpić później już w trakcie eksploatacji, chociaż często nie jest w ogóle potrzebna.

Język zapytań będący podzbiorem 4GL jest sprzężony z mechanizmem obsługi ekranu, co istotnie ułatwia projektowanie przeglądów. Forma zapytań zbliżona jest do naturalnego języka angielskiego. Użycie indeksów podczas wyszukiwania danych następuje automatycznie, chociaż istnieje możliwość wyboru konkretnego indeksu, którym w danym zapytaniu chcemy się posłużyć. Jeżeli indeks dla danego klucza nie został zdefiniowany, wyszukanie danej jest w dalszym ciągu możliwe, choć czasochłonne.

Ponieważ aplikacje muszą być dopasowane do różnych potrzeb i przyzwyczajeń użytkowników, PROGRESS 4GL posiada wygodny mechanizm kontroli klawiatury. Umożliwia on narzucenie własnego schematu obsługi w każdym z interakcyjnych elementów języka.

Najważniejszą cechą PROGRESS-a jest jego kompletność. Oznacza to możliwość stworzenia całości aplikacji w 4GL bez konieczności odwoływania się do języków trzeciego poziomu.

### b) PROGRESS SQL

PROGRESS SQL jest językiem obsługi relacyjnych baz danych opartym na standardzie SQL86 ANSI. Spełnia on wszystkie wymagania pierwszego poziomu oraz większość wymagań poziomu drugiego standardu. Dodatkowo język ten udostępnia wiele funkcji charakterystycznych dla PROGRESS-a.

W przeciwieństwie do innych istniejących implementacji SQL język ten zanurzony jest w 4GL, a więc nie wymaga korzystania z języków trzeciej generacji. Instrukcje SQL mogą być przeplatane z kodem 4GL.

Rozszerzenia w stosunku do standardu SQL86 stanowią polecenia ALTER TABLE, DROP TABLE, DROP VIEW, CREATE INDEX, DROP INDEX, REVOKE, charakterystyczne dla PROGRESS-a typy danych DATE i LOGICAL, a także frazy definicji formatu danych i formatki ekranowej. Definicje te są wspólne dla PROGRESS SQL oraz 4GL.

### c) Wykorzystanie języków trzeciej generacji (C, COBOL, PASCAL)

Mimo iż w większości zastosowań nie zachodzi potrzeba użycia języków trzeciej generacji, to jednak możliwość taka istnieje i jest wykorzystywana wszędzie tam, gdzie PROGRESS pracuje w nietypowym otoczeniu.

Istnieją dwa interfejsy służące do komunikacji PROGRESS-a z 3GL<sup>12</sup>. Pierwszy z nich - HLC<sup>13</sup> - pozwala na wywołanie procedur napisanych w C z poziomu 4GL. Daje to możliwość zdefiniowania nietypowych funkcji numerycznych, bezpośredniej obsługi urządzeń zewnętrznych jak n.p. sprzęt video, czy przetworniki przemysłowe, a także umożliwia bezpośrednie operacje na danych innych programów.

Drugi interfejs - HLI<sup>14</sup> - umożliwia umieszczenie zapytań PROGRESS SQL w kodzie języka niskiego poziomu (C, COBOL-u, Pascala)<sup>15</sup>. Daje to możliwość wykorzystania mechanizmu zarządzania bazą danych PROGRESS-a z poziomu 3GL. Oznacza to nie tylko dostęp do danych, ale również pełne ich bezpieczeństwo zapewnione przez RDBMS. Ważnym elementem jest możliwość dynamicznego formułowania zapytań SQL w trakcie pracy programu 3GL.

## 5. Mechanizmy pracy współbieżnej

W środowisku wielodostępnym, podczas równoczesnej pracy wielu użytkowników, występują dwa problemy. Pierwszy z nich to powstawanie "wąskich gardeł", co prowadzi do bardzo szybkiego spadku wydajności systemu wraz ze wzrostem liczby aktywnych klientów. W przypadku PROGRESS-a antidotum stanowi architektura CLIENT/SERVER oraz zastosowanie przetwarzania wielotorowego, które szerzej omówiono w p. 7.

Znacznie bardziej złożony jest problem współdzielenia zasobów. PROGRESS udostępnia całkowicie automatyczny mechanizm rezerwacji zasobów na poziomie pojedynczych rekordów lub całych plików. W zależności od rodzaju wykonywanej operacji rezerwacja dokonywana jest w trybie wyłącznym (EXCLUSIVE LOCKING) lub w trybie dzielonego dostępu (SHARED LOCKING). Sterowanie rezerwacją zasobów możliwe jest przez odpowiednie konstruowanie bloków w procedurach.

## 6. Mechanizmy ochrony danych

### a) Ochrona przed niepożądanym dostępem

W niektórych zastosowaniach, jak n.p. bankowość, czy wojskowość szczególnie ważna staje się ochrona danych przed niepożądanym dostępem. PROGRESS wyposażony jest we wbudowany system ochrony danych umożliwiający utworzenie struktury użytkowników weryfikowanych za pomocą indywidualnych haseł, nadanie uprawnień administratora systemu wyodrębnionej podgrupie użytkowników, a także zdefiniowanie zakresu dostępu do danych dla każdego użytkownika. Jeżeli struktura taka została zdefiniowana, system automatycznie weryfikuje tożsamość użytkownika w momencie startu, a następnie sprawdza uprawnienia przy każdej próbie dostępu do danych.

Za pomocą omówionego mechanizmu administrator systemu może zezwolić, bądź zabronić jednego z czterech (Can-Read, Can-Write, Can-Create, Can-Delete) typów dostępu na poziomie plików, a także jednego z dwóch typów dostępu (Can-Read, Can-Write) na poziomie pól rekordu.

Jeżeli twórca aplikacji zamierza wprowadzić bardziej złożone mechanizmy kontroli dostępu (n.p. blokada pewnych operacji, a nie tylko dostępu do danych), może dokonać tego, wykorzystując istniejące w 4GL funkcje CAN-DO oraz USERID.

---

<sup>12</sup> skr. ang. 3rd Generation Language - język trzeciej generacji.

<sup>13</sup> skr. ang. Host Language Call.

<sup>14</sup> skr. ang. Host Language Interface.

<sup>15</sup> dostępność poszczególnych języków zależy od używanego sprzętu.



#### b) Zapewnienie spójności logicznej podczas wprowadzania

Bardzo rzadko zdarza się, że wprowadzana dana jest całkowicie niezależna od pozostałej informacji zgromadzonej w systemie. Najczęściej dane uzupełniają się nawzajem i są sensowne tylko w pewnych konfiguracjach (nie ma sensu n.p. rachunek bez właściciela, czy nr telefonu bez przypisanego abonenta). Również wartość danych zazwyczaj nie może być dowolna. Ponieważ zawsze najbardziej zawodnym elementem systemu jest człowiek, sensowność danych należy sprawdzać przede wszystkim podczas ich wprowadzania, lub modyfikacji.

Do tego celu służą w PROGRESS-ie klauzule walidacji. Są to często złożone warunki logiczne sprawdzane automatycznie podczas edycji pola lub zmiennej. W odniesieniu do pól bazy są one definiowane łącznie ze strukturą bazy w Data Dictionary. Warunek walidacji zmiennej możemy zadeklarować w momencie deklaracji samej zmiennej. Jeżeli warunek walidacji pola lub zmiennej nie będzie spełniony podczas edycji, użytkownik nie będzie mógł opuścić edytowanego pola, chyba że poprawi błąd lub zrezygnuje z edycji, co jest jednoznaczne z zachowaniem starej wartości.

Do niespójności danych może doprowadzić również usunięcie części z nich, jeżeli do usuniętych rekordów odnoszą się dane zgromadzone w innej części bazy. Przed takim działaniem można zabezpieczyć się w PROGRESS-ie, definiując w opisie pliku warunki konieczne do spełnienia w momencie usuwania rekordów z pliku.

Zarówno w pierwszym, jak i drugim przypadku istnieje możliwość sformułowania komunikatu, który będzie automatycznie wyświetlany w przypadku stwierdzenia nieprawidłowości.

#### c) Ochrona przed uszkodzeniem lub utratą danych

W dużych systemach istnieje realne niebezpieczeństwo utraty części, a nawet całości danych, na skutek awarii sprzętu lub systemu operacyjnego. Pomysłowość materii nieożywionej wydaje się być tutaj nieskończona, tak że trudno spodziewać się, aby problem ten został kiedykolwiek definitywnie rozwiązany. Minimalizacja prawdopodobieństwa utraty danych, bądź zaistnienia w nich przekłamań (często gorszych w skutkach niż całkowita utrata danych), zależy od ilości i różnorodności zastosowanych zabezpieczeń. Mechanizmy zaimplementowane w systemie PROGRESS obejmują:

##### **Two-phase commit<sup>16</sup>**

Przy odwołaniach do wielu baz w środowisku rozproszonym spójność danych gwarantowana jest przez dwufazowy protokół potwierdzeń. Obejmuje on fazę potwierdzenia zapisu przez poszczególne serwery. Jeżeli z jakichkolwiek powodów część danych nie została zapisana, całość operacji jest anulowana.

##### **Before-imaging<sup>17</sup>, transakcje**

Jeżeli przechowywana w bazie informacja ma być spójna, przy każdej operacji modyfikującej dane powinna obowiązywać zasada "wszystko albo nic". Zabezpieczenie przed częściową, niekompletną modyfikacją stanu bazy osiągane jest w PROGRESS-ie poprzez zastosowanie mechanizmu transakcji. W momencie rozpoczęcia transakcji bieżący stan modyfikowanej części bazy zapamiętywany jest w specjalnym pliku \*.bi (tzw. before-image file). Jeżeli z jakichkolwiek powodów (n.p. zaniku zasilania, błędu systemu lub zaniechania operacji przez użytkownika) transakcja nie zostanie pomyślnie zakończona, system gwarantuje odtworzenie stanu bazy sprzed rozpoczęcia transakcji.

W systemie PROGRESS możliwe jest zagnieżdżanie transakcji. Dzięki temu możliwe jest częściowe wycofanie dokonanych już zmian w bazie, jeżeli w jednym z kolejnych kroków podczas wprowadzania (lub usuwania) większej porcji danych będziemy chcieli wycofać ostatnią operację, nie wycofując równocześnie poprzednich. W przypadku awarii systemu odtwarzanie stanu bazy obejmie oczywiście najbardziej zewnętrzną transakcję.

<sup>16</sup> ang. dwufazowy protokół potwierdzeń.

<sup>17</sup> ang. przechowanie obrazu pierwotnego.

### **After-imaging<sup>18</sup>, Roll Forward Recovery<sup>19</sup>**

O ile system transakcji oraz przechowywania obrazu pierwotnego zabezpiecza użytkownika przed utratą danych na skutek awarii systemu, o tyle jest on bezradny w przypadku fizycznego uszkodzenia nośnika. Jeżeli zawarte w bazie są szczególnie ważne, wówczas możemy zlecić systemowi oddzielne zapamiętywanie po zakończeniu transakcji wszystkich zmian, które zostały w niej wprowadzone. Mechanizm ten zwany after-imaging nie jest automatyczny. Wymaga on zadeklarowania lokalizacji pliku \*.ai z obrazem końcowym w jednej z opcji startowych systemu. Idealnym rozwiązaniem jest lokalizacja plików \*.ai oraz \*.bi na różnych dyskach, ponieważ prawdopodobieństwo utraty obydwu plików jednocześnie jest wtedy minimalne.

Wykorzystanie mechanizmu after-image wiąże się z wprowadzeniem procedury regularnego składowania stanu bazy. W przypadku awarii nośnika należy odtworzyć stan bazy z ostatniego pełnego składowania, a następnie, wykorzystując odpowiednie narzędzia oraz plik \*.ai, doprowadzić bazę do stanu bezpośrednio poprzedzającego awarię. Istotne jest ustalenie właściwego protokołu składowania, ponieważ nadmierny rozrost pliku \*.ai może być przyczyną nieprzewidzianych kłopotów (n.p. zapelnienie dysku zawierającego ten plik).

Proces odtwarzania stanu bazy na podstawie zapisów kolejnych modyfikacji nazywany jest odtwarzaniem sukcesywnym (Roll Forward Recovery).

### **Składowanie stanu bazy**

System PROGRESS dostarcza bardzo wielu możliwości składowania stanu bazy. Służy do tego narzędzie PROGRESS Backup. Umożliwia ono dokonanie składowania w tle (online backup) bez konieczności przerywania pracy systemu. Możliwości PROGRESS Backup zależą od systemu operacyjnego i trybu użytkowania bazy. W systemach DOS, OS/2 oraz BTOS/CTOS możliwe jest dokonanie pełnego bądź przyrostowego<sup>20</sup> składowania na dwa rodzaje nośników:

- dyskietki,
- pojedynczy zbiór na dysku twardym.

W systemach DOS i OS/2 można przenieść taki zbiór na taśmę za pomocą narzędzi systemowych. Bazy pracujące pod kontrolą systemów UNIX lub VMS mogą być bezpośrednio składowane na jeden z następujących nośników:

- dyskietki,
- taśma 9-cio ścieżkowa,
- taśmę kasetową,
- dysk kasetowy,
- niezależne urządzenie dyskowe,
- oddzielny plik w obrębie systemu,
- oddzielny plik pozasystemowy.

Nie jest możliwe dokonanie składowania w tle w systemach pracujących bez podziału pamięci (jak DOS, czy BTOS) oraz w przypadku, gdy baza użytkowana jest w trybie jednodostępnym (single-user mode). Jeżeli baza jest dużych rozmiarów, mamy możliwość dokonania składowania przyrostowego. Zamiast całości bazy zapamiętywana jest wówczas jedynie informacja z tych jej sektorów, które uległy modyfikacji. Dzięki temu mechanizmowi pełne składowanie może odbywać się raz na tydzień lub miesiąc, w zależności od intensywności zmian w bazie.

## **7. Mechanizmy wewnętrzne, a efektywność systemu**

### **Architektura CLIENT/SERVER**

Z myślą o przetwarzaniu danych w środowisku sieciowym projektanci systemu już od pierwszych wersji rozdzielili funkcje serwera bazy danych od procesów klienta. Proces klienta komunikuje się z użytkownikiem, obsługując stronę zewnętrzną aplikacji oraz interfejs

---

<sup>18</sup> ang. zabezpieczenie obrazu końcowego.

<sup>19</sup> ang. odtwarzanie sukcesywne.

<sup>20</sup> ang. incremental backup.



użytkownika. Proces serwera zarządza bazą danych, kontrolując dostęp i modyfikacje danych. Technika ta znana pod nazwą cooperative processing<sup>21</sup> daje znaczące korzyści czasowe zauważalne w dużych, wieloużytkownikowych systemach.

### **Przetwarzanie wielotorowe (MULTI-THREADED/MULTI-SERVER processing)**

Rozwiązaniem zwiększającym wydajność systemu w środowisku rozległym jest zastosowanie przetwarzania wielotorowego na poziomie serwerów<sup>22</sup>. Każdy z lokalnych procesów klientów komunikuje się z bazą poprzez własny proces serwera. Dzięki temu ograniczana jest komunikacja międzyprocesowa. Serwery obsługujące zapytania odległe komunikują się zazwyczaj z wieloma procesami klientów. Aby nie dopuścić do powstania wąskiego gardła, którym byłby z pewnością przeciążony serwer, wprowadzony został dodatkowy proces administrujący (ang. broker), który rozdziela nadchodzące zgłoszenia pomiędzy najmniej obciążone procesy serwerów. Wraz ze wzrostem obciążenia systemu (włączaniem się nowych użytkowników) broker uruchamia nowe procesy serwerów, starając się utrzymać jednakowy czas średni odpowiedzi systemu na zapytanie.

### **Zmienna długość rekordu**

Zapamiętywanie danych ze zmienną długością jest nie tylko wygodne z punktu widzenia programisty. Podstawowa korzyść płynąca z zastosowania tej techniki to oszczędność miejsca na dysku dochodząca w niektórych systemach do 60%. Rozmiar danych dyskowych ma duży wpływ na szybkość działania systemu, a więc uzyskiwane są również korzyści czasowe.

### **Mechanizm indeksów**

Indeksacja zawartości pliku według pewnego klucza jest świadomym działaniem programisty. Zadaniem indeksu jest nie tylko organizacja zgromadzonej informacji w zadanym porządku, ale również zapewnienie jak najkrótszego czasu wyzukiwania danych. Efektywność tej techniki zależy w dużej mierze od wiedzy programisty o sposobach wykorzystania projektowanego systemu przez jego przyszłych użytkowników, a także od umiejętności prawidłowego wykorzystania tej wiedzy. Nawet jednak najlepiej zaprojektowana struktura indeksów jest tylko na tyle wydajna, na ile wydajny jest algorytm indeksacji. W systemie PROGRESS wykorzystano metodę znaną jako skondensowane wielopolowe B-drzewo<sup>23</sup>. Jest to bardzo elastyczny i wydajny algorytm, mający zdecydowany wpływ na ogólną efektywność systemu.

## **8. Polska wersja językowa**

PROGRESS Software Corporation stara się dostosowywać dostarczane oprogramowanie do wymogów tych krajów, w których jest ono dostępne. System PROGRESS operuje na rozszerzonym zestawie znaków ASCII, a więc pozwala na posługiwanie się alfabetem narodowym. Istnieje również możliwość zdefiniowania własnych reguł kolejności znaków alfabetu, co jest niezbędne dla pełnej użyteczności systemu.

Zostały opracowane również specjalne wersje PROGRESS-a dedykowane dla poszczególnych krajów. W chwili obecnej istnieją wersje: holenderska, francuska, włoska, szwajcarska, angielska, niemiecka, hiszpańska, duńska, norweska, szwedzka oraz fińska. Oprócz z góry przygotowanych tablic kolejności znaków są one wyposażone w pełen zestaw komunikatów systemowych w danym języku. Uwzględniają również specyficzne dla danego kraju reguły pisania daty, godziny, czy liczb dziesiętnych.

Opracowanie każdej wersji językowej to olbrzymia praca. W przypadku Polski najważniejszą jej część polegającą na przetłumaczeniu ok. 2000 komunikatów systemowych (wraz z kilkumierszowymi objaśnieniami) została już wykonana. Ponieważ już od listopada 1991 PROGRESS daje możliwość deklaracji własnych tablic sortowania i konwersji, w chwili obecnej

---

<sup>21</sup> Brak jest jednoznacznego odpowiednika polskiego tego terminu.

<sup>22</sup> ang. Multi-Threaded Multi-Server processing.

<sup>23</sup> ang. compressed multi-field B-tree indexing.

zaspokaja on większość potrzeb polskiego użytkownika. Pełna polska wersja językowa, która jest przygotowywana przez PSC wraz z CSBI obejmuje dodatkowo przetłumaczenie wszystkich programowych elementów środowiska PROGRESS-a (Data Dictionary, Fast Track, Results), a także włączenie Polski, jako kraju wybieranego opcją startową systemu (o ile uda się do tego czasu stworzyć polski standard tablicy kodowej).

#### IV. Korzystanie z systemu

##### 1. Projektowanie struktur danych

Pierwszą fazą w tworzeniu nowej aplikacji jest zaprojektowanie struktur danych. Informacja zawarta w bazie powinna być kompletna. Od umiejętnego jej pogrupowania oraz poindeksowania zależy wydajność systemu. W krańcowym przypadku brak logiki w strukturach danych może doprowadzić do całkowitej bezużyteczności systemu, mimo uwzględnienia wszystkich niezbędnych informacji.

W systemie PROGRESS definiowanie struktur danych odbywa się w trakcie interaktywnej pracy ze słownikiem danych. Możliwe jest również wcześniejsze przygotowanie takich definicji i wprowadzenie ich za pomocą opcji LOAD Data Definitions z menu Admin w Data Dictionary. Jeszcze innym sposobem jest wykorzystanie komend DDL<sup>24</sup> zawartych w oferowanym przez PROGRESS standardzie SQL. Podstawową metodą pozostaje jednak bezpośrednia praca ze słownikiem danych.

W przypadku konieczności zmiany istniejącej definicji można operację taką przeprowadzić w dowolnym momencie, również na wypełnionej danymi bazie. Należy jednak przy tym pamiętać, że nie jest dozwolona zmiana typu danych.

##### 2. Budowa aplikacji prototypowej

Mając przygotowane struktury danych, możemy przystąpić do budowy funkcjonalnego szkieletu aplikacji. Bardzo pomocnym narzędziem jest tutaj FAST TRACK. Za pomocą edytora menu możemy stworzyć niemal dowolną strukturę menu. Poszczególnym punktom menu przypisujemy już istniejące, bądź dopiero planowane funkcje. Menu tworzone za pomocą FAST TRACK-a można w dowolnej chwili przeprojektować, a równocześnie jest ono cały czas gotowe do wykorzystania.

Operacje przeprowadzane na bazie danych wiążą się zazwyczaj z konkretnymi formatkami ekranowymi. Wbudowany w FAST TRACK-a edytor formatów ekranowych daje nam możliwość projektowania zawartości ekranu w sposób interaktywny. Jest to bardzo wygodne, ponieważ zwalnia programistę od konieczności jawnego deklarowania dziesiątków szczegółów niezbędnych do opisanego formatki. Ma to oczywisty wpływ na tempo pracy nad aplikacją.

Jeżeli operacje związane ze zdefiniowaną uprzednio formatką mieszczą się w standardowym zestawie obejmującym przegląd, modyfikację, dopisanie i usunięcie danych, możemy wykorzystać generator procedur QBF.

Inną standardową czynnością wykonywaną często w aplikacjach jest generacja raportów. Tu również FAST TRACK daje nam narzędzie umożliwiające szybkie, interaktywne definiowanie struktury raportu. Możemy niezależnie definiować stronę tytułową, nagłówki i stopki poszczególnych stron, podsumowania częściowe i całościowe raportu, a nawet tworzyć raporty zagnieżdżone, oparte o dane z plików powiązanych relacją 1-n.

W ten sposób, bez pisania pojedynczej linii kodu, możemy stworzyć prototypowe środowisko

---

<sup>24</sup> skr. ang. Data Definition Language - język definicji danych.



pracy użytkownika, dające mu możliwość wykonywania większości operacji dostępnych w kompletnym systemie. Niektóre fragmenty aplikacji, a w szczególności procedury przetwarzające dane w sposób niewidoczny dla użytkownika, muszą być napisane w 4GL. Edytor menu umożliwia włączenie takich procedur do prototypu, który w tym momencie może stać się wersją docelową. Z biegiem czasu możliwe jest zastępowanie poszczególnych fragmentów prototypu odpowiednimi procedurami języka 4GL, dla wielu użytkowników jest to jednak nieopłacalne.

### 3. Tworzenie aplikacji w 4GL

Chociaż wykorzystanie FAST TRACK-a daje duże korzyści czasowe i organizacyjne, to jednak niektóre zastosowania wymagają napisania dużej części lub nawet całości oprogramowania w języku PROGRESS 4GL. Zmuszać do tego mogą szczególne wymagania dotyczące wyglądu lub obsługi menu, zastosowanie nietypowych trybów edycji lub wiele innych powodów.

#### PROGRESS Editor

W języku PROGRESS 4GL pisze się, wykorzystując wbudowany edytor PROGRESS-a lub dowolny inny edytor plików ASCII dostępny w danym systemie operacyjnym. Edytor wbudowany umożliwia nam stałą kontrolę poprawności programu. Procedura może zostać w każdej chwili uruchomiona, zaś jakikolwiek błąd kompilacji bądź wykonania jest natychmiast raportowany. Edytor wyposażony jest we wszystkie standardowe funkcje z wyszukiwaniem oraz kopiowaniem fragmentów tekstu włącznie. Akcje przypisane poszczególnym klawiszom mogą początkowo wydać się zbyt dziwnione, jednak ich niezaprzeczalną zaletą jest niezależność od systemu operacyjnego. Edytor PROGRESS-a pracuje wszędzie tak samo, a więc przyzwyczajając się trzeba tylko raz.

#### PROGRESS Help

Klawisz F2 służy w PROGRESS-ie do wywoływania pomocnika. Dotyczy to zarówno środowiska pracy programisty (edytora), jak i gotowych aplikacji, gdzie przypisanie funkcji pomocy klawiszowi F2 jest zalecane, choć nie obowiązkowe. W środowisku edytora naciśnięcie klawisza pomocy powoduje pojawienie się menu z zestawem kilkunastu funkcji. Można wówczas skorzystać ze skróconych skrótów, zawierającego poszerzone opisy wszystkich możliwych błędnych sytuacji, zasięgnąć informacji na temat składni języka, dostępnych funkcji, operatorów i zastrzeżonych słów kluczowych, przejrzeć listę wszystkich ograniczeń PROGRESS-a, przypomnieć sobie funkcje poszczególnych klawiszy, przejść do Data Dictionary lub modułu bibliotekarza, obejrzeć zawartość bieżącej kartoteki systemowej, wyjść do systemu operacyjnego, a także zakończyć sesję.

#### Korzystanie z bibliotek gotowych rozwiązań

W języku PROGRESS procedury mogą być przechowywane w postaci źródłowej, bądź skompilowanej do tzw. r-kodu. Procedury skompilowane można z kolei umieszczać w zbiorach bibliotecznych, co nie tylko porządkuje pracę, ale również przyspiesza wykonanie programu.

Wraz z systemem dostarczane są biblioteki procedur spakowanych w postaci źródłowej. Procedury te napisane przez doświadczonych programistów PSC można włączyć do własnej aplikacji lub posłużyć się nimi jako wzorem przy pisaniu własnych wersji. Procedury przykładowe mają charakter uniwersalny i obejmują większość "stałych fragmentów gry", z którymi styka się programista podczas pracy nad aplikacją. Należą do nich różne formaty menu, procedury wyboru, przeglądów, obliczeń matematycznych i.t.p. W niektórych przypadkach mogą one stanowić alternatywę do funkcji FAST TRACK-a.

#### Korzystanie z generatorów programu

Działanie FAST TRACK-a polega na wykorzystaniu procedur uniwersalnych oraz danych zgromadzonych w specjalnie zmodyfikowanej strukturze bazy. Dzięki temu aplikacja FAST TRACK-owa jest spójna i łatwiejsza do kontroli. Jednakże na specjalne życzenie programisty FAST TRACK jest w stanie wygenerować fragment programu w 4GL ekwiwalentny w stosunku do dowolnej ze swoich funkcji. Fragmenty te, bezpośrednio lub po modyfikacji, można wykorzystać w tekście własnej aplikacji.

Również w Data Dictionary istnieje możliwość skorzystania z generatora fragmentów kodu. Generuje on gotowe pliki włączane zawierające złożone instrukcje ASSIGN, FORM oraz

## DEFINE WORKFILE.

### Wykorzystanie raportów słownika danych i kompilatora

Podczas pracy nad aplikacją ważnym elementem jest zapewnienie możliwości przeglądu wszystkich zdefiniowanych struktur. Do tego celu służy podmenu Reports w Data Dictionary.

Inne możliwości raportowania, również istotne dla programisty, ma kompilator PROGRESS-a. Dają je opcje LISTING oraz XREF komendy COMPILE. Pierwsza z nich generuje kompletny listing programu zawierający:

- nazwę kompilowanego pliku,
- nazwę bazy danych,
- datę i godzinę kompilacji,
- numer każdej linii procedury,
- numer bloku, do którego należy każda instrukcja,
- kompletny tekst wszystkich plików włączonych,
- pełne nazwy wszystkich podprocedur.

Opcja XREF generuje listę powiązań<sup>25</sup> pomiędzy obiektami procedur i baz danych. Każda linia opisuje jedno powiązanie za pomocą następujących pięciu elementów:

- nazwa procedury,
- nazwa pliku (procedura może być w kilku),
- numer linii,
- typ powiązania,
- identyfikator obiektu.

Przy dużym rozmiarze aplikacji korzystanie z wymienionych możliwości raportowania jest bardzo pomocne, a czasami nawet niezbędne.

## V. Plany rozwoju systemu

Zazwyczaj, jeżeli jakaś firma ogłasza swoje zamierzenia, traktujemy to niepoważnie. Rzeczywiście, w większości przypadków realizacja planów nie następuje zgodnie z podanym harmonogramem. Niektóre z planów pozostają na zawsze w sferze zamierzeń. PROGRESS Software Corporation jest w tej dziedzinie chlubnym wyjątkiem. Do swoich zobowiązań podchodzi bardzo poważnie, a kolejne wersje produktów wydawane są z regularnością szwajcarskiego zegarka.

Wersje produktów PSC oznaczane są numerem oraz dużą literą łacińską. Każdy kolejny numer oznacza nową generację produktów o rozszerzonych cechach użytkowych. Równocześnie poszerzana jest lista dostępnych platform sprzętowych.

Ponieważ realizacja wszystkich założeń nowej generacji produktu jest procesem długotrwałym, jest ona dzielona na szereg etapów, w których opracowywane są kolejne wersje wewnątrz danej generacji. Wersje te, oznaczane literami, wzbogacane są o kolejne cechy, aż do całkowitej realizacji pierwotnego projektu. W ramach tych wersji mogą również zaistnieć pewne rozszerzenia sprzętowe nie wychodzące jednak zazwyczaj poza już oprogramowane klasy sprzętu.

Ze względu na stały kontakt z użytkownikami plany rozwoju systemu są cały czas dopasowywane do bieżących potrzeb rynku. Przykładem elastyczności firmy może być tutaj błyskawiczna reakcja na wzrost popularności systemów okienkowych (patrz p. III.2.d).

Poniżej wymieniono główne kierunki rozwoju systemu PROGRESS.

### Opracowanie zaawansowanego środowiska CASE

Narzędzia CASE nie są zwykłymi generatorami aplikacji, chociaż generacja kodu może być jedną z funkcji środowiska CASE. Zastosowanie narzędzi tego rodzaju wiąże się przede wszystkim z obroną metodologią tworzenia oprogramowania i wspomaga wszystkie jej etapy. W chwili obecnej PROGRESS nie posiada własnego środowiska CASE w pełnym tego słowa znaczeniu.

---

<sup>25</sup> ang. cross reference list.



Jednakże pozwala na wykorzystanie istniejących i uznanych w świecie środowisk Excelerator i Knowledgware. W najbliższym czasie spodziewana jest też premiera narzędzia CASE powstałego w całości w PSC. Gwarancją jakości tego narzędzia ma być nazwisko kierownika tego projektu - Elliota Chikofskiego - jednego z najlepszych na świecie specjalistów w tej dziedzinie.

#### **Ewolucja w kierunku systemu wielojęzycznego**

Już dzisiaj dostępnych jest kilkanaście wersji narodowych PROGRESS-a. W wersjach tych wszystkie komunikaty systemowe wyświetlane są w języku użytkownika. Charakterystyczne dla danego języka są również reguły kolejności znaków, sposób pisania daty, godziny oraz liczb dziesiętnych. W wersji 7 PROGRESS-a przewidziano mechanizmy dynamicznego przełączania wersji językowej w trakcie pracy systemu. Umożliwi to zastosowanie PROGRESS-a wszędzie tam, gdzie potencjalni użytkownicy są różnych narodowości, a więc w obsłudze ruchu turystycznego, kompaniach wielonarodowych i.t.p. Wprowadzenie tej możliwości wiąże się z modyfikacją struktury bazy, w której każdy identyfikator, etykieta czy komunikat będzie deklarowany w kilkunastu wersjach.

#### **Zapewnienie bezpośredniej przenośności oprogramowania**

W wersji 6 PROGRESS-a przeniesienie aplikacji w nowe środowisko wiąże się z koniecznością jej rekompilacji. W następnych wersjach problem ten przestanie istnieć, ponieważ r-kod generowany w trakcie kompilacji będzie całkowicie niezależny od sprzętu.

#### **Nowe bramy do baz innych systemów**

Do czasu ukazania się wersji siódmej przewiduje się udostępnienie możliwości bezpośredniej komunikacji z bazami SYBASE oraz DB2. Trwają również prace nad bramami do INFORMIX-a i INGRESS-a.

## **VL    Od autora**

Niniejszy tekst opracowałem na bazie materiałów dostarczonych przez firmę PROGRESS Software Corporation. Nie jest ono z pewnością dokumentacją techniczną, omawia jednak większość elementów systemu dając czytelnikowi możliwość wyrobienia sobie poglądu zarówno na temat firmy, jak i jej produktu.

Oryginalna dokumentacja firmowa liczy ponad 4000 stron (13 podręczników). Do tego dochodzą publikacje uzupełniające, biuletyny techniczne, marketingowe, czasopisma i inne materiały związane z systemem. Zgromadzonych w nich informacji nie da się przedstawić w tak krótkim opracowaniu nie pomijając pewnych szczegółów, czy nawet całych tematów. Tych, którzy zainteresowani są uzyskaniem odpowiedzi na bardziej szczegółowe pytania wypada odesłać do firmy CSBI, która dysponuje pełnym zestawem materiałów firmowych. Jej pracownicy mogą również służyć osobistym doświadczeniem nabytym podczas trzyletniej już pracy z systemem.

W opracowaniu świadomie nie podjąłem próby porównania systemu PROGRESS z jego konkurentami. Każdy system posiada swoje słabe strony, a ich wypominanie jest moim zdaniem przejawem braku szacunku dla dziesiątek programistów tracących swój czas i zdrowie przy pracy nad kolejnymi wersjami produktu. Nie podejmując krytyki systemów konkurencyjnych chciałbym jednak wyrazić przekonanie, że to właśnie PROGRESS, a nie ORACLE czy INFORMIX, jest systemem najlepiej dostosowanym do polskich warunków. Myślę, że po zapoznaniu się z niniejszą charakterystyką, a szczególnie jej częścią omawiającą przenośność systemu i różnorodność platform sprzętowych, przekonania takiego nabierze również czytelnik.

METODY  
ANALIZY I PROJEKTOWANIA SYSTEMÓW  
W PRAKTYCE

Stanisław Wrycza  
Uniwersytet Gdański  
Katedra OPD  
81-824 Sopot  
Armi Krajowej 119/121  
Tel. (58) 51 00 61 w. 400

1. WPROWADZENIE.

Metody i techniki analizy i projektowania systemów które w pierwszej połowie lat osiemdziesiątych były przedmiotem rozważań na konferencjach naukowych znajdują obecnie coraz szersze zastosowanie p r a k t y c z n e. Można mówić nawet o przyjęciu pewnych standardów z tej dziedziny w krajach o wysokiej technologii informatycznej.

C e l e m niniejszego referatu jest wprowadzenie do całodziennych warsztatów ukierunkowanych na praktyczne zastosowanie metod i technik wstępnych faz cyklu życia systemu tj. planowania, analizy i projektowania. Warsztaty te opierają się na rozwiązaniu przedstawionego w załączniku przypadku.

Stosownie do celu warsztatów w drugiej części referatu przedstawiono panoramę metod i technik tworzenia systemów informatycznych. Prezentację tą powiązano z fazami c y k l u życia systemu. Szczególną uwagę zwrócono tu na stosunkowo mało znane w Polsce metody planowania systemu. Trzecia część zawiera szczegółowy harmonogram warsztatów.



## 2. METODY I TECHNIKI TWORZENIA SYSTEMÓW W CYKLU ŻYCIA SYSTEMU.

Dobór i użytkowanie metod, technik i narzędzi tworzenia systemów informatycznych należy powiązać ściśle z cyklem życia systemu. Istnieją kontrowersje co do zakresu, kolejności faz i ich nazewnictwa w odniesieniu do procesu TSI. Dotychczasowe próby standaryzacyjne jak dotąd nie uzyskały powszechnej akceptacji. Ogólnie można przyjąć, iż cykl życia systemu obejmuje następujące fazy:

- planowanie,
- analizę,
- projektowanie,
- wdrożenie,
- użytkowanie i kontrolę.

Każdej z tych faz odpowiadają specyficzne metody i techniki, choć w niektórych przypadkach służą one kilku fazom. W szczególności dotyczy to faz analizy i projektowania.

O ile metody analizy i projektowania torują sobie w kraju powoli zrozumienie o tyle metody planowania systemu informacyjnego są zupełnie zaniedbywane i niedoceniane. Tymczasem związana z nimi dziedzina Information Planning rozwija się niezwykle dynamicznie. W konkretnych realizowanych projektach chodzi tu o osiągnięcie dwu celów wzajemnie ze sobą powiązanych:

- stworzenie systemów informatycznych skutecznie wspomagających strategiczne cele firmy,
- osiągnięcie zaangażowania (pośrednio więc i odpowiedzialności) kierownictwa firmy w proces TSI.

Istnieje cały szereg aktywizujących, heurystycznych metod identyfikowania przez kierownictwo obszarów działalności firmy, których informatyzacja sprzyjać będzie rozwojowi firmy, skutecznemu wdrażaniu jej strategii rynkowej oraz podnoszeniu jej konkurencyjności. Wśród metod tych w pierwszym rzędzie należy wymienić:

- sesję Metaplan.
- Analizę Istotnych Czynników Powodzenia.
- model spójności Broekstry.

Szczegółowa charakterystyka każdej z tych metod jest zagadnieniem samym w sobie. Warto jednak podać nawet krótkie omówienie każdej z tych metod, choćby, z tego powodu, że są one w Polsce mało znane.

Bardzo efektywnym sposobem planowania systemu informacyjnego jest sesja M e t a p l a n u. Założenia jej są stosunkowo proste, lecz jak wskazuje doświadczenie, osiąga się bardzo użyteczne rezultaty oraz wysokie zaangażowanie kierownictwa. W następujących po sobie "taktach", metodą dyskusji, ustalania priorytetów i wreszcie prezentacji określa się kolejno:

- cele firmy,
- zagrożenia,
- działania dla osiągnięcia celów,
- działania dla uniknięcia zagrożeń,
- specyfikację systemów informacyjnych wspomagających osiąganie celów i pozwalających na uniknięcie zagrożeń.

Pomijamy tu szczegółowe, aktywizujące, często atrakcyjne techniki realizacji sesji Metaplan.

Kolejną metodą planowania systemów informacyjnych jest metoda Istotnych C z y n n i k ó w Powodzenia (Critical Success Factors). Oznacza ona wyodrębnienie pięciu do ośmiu obszarów działalności gospodarczej firmy, w których pozytywny rezultat gwarantuje jej pomyślne funkcjonowanie na rynku. Istotnych Czynników Powodzenia używa się do wydzielenia przez kierownictwo firmy głównych obszarów zainteresowania dla zapewnienia ciągłości i rozwoju jej funkcjonowania. Identyfikacja Istotnych Czynników Powodzenia następuje poprzez:

- przeprowadzenie warsztatu pozwalającego określić cele i priorytety firmy,
- opracowanie i przeprowadzenie ankiety wśród kierownictwa firmy na bazie rezultatów warsztatów.



- po zapoznaniu się członków kierownictwa z wynikami ankiet przeprowadzenie kolejnego warsztatu do ustalenia ostatecznej listy Istotnych Czynn timer Powodzenia.

Metoda ta ma bezpośredni związek z dalszymi fazami cyklu życia systemu. Mianowicie w wyniku analizy scenariuszy decyzji można dokonać wyboru systemów priorytetowych. Służą one z kolei opracowaniu ogólnych projektów systemów metodą prototypowania.

Interesującym sposobem inicjowania informatyzacji firmy jest model s p ó j n o ś c i Broekstry. Jest on ukierunkowany na analizę wpływu zmian dokonanych w jednej sferze działania na inne sfery. Broekstra wyróżnia pięć podstawowych czynników określających funkcjonowanie firmy i jej miejsce na rynku. Są to:

- kombinacja produkt/rynek,
- technologia,
- organizacja,
- kadry,
- dominująca koalicja czyli formalne i nieformalne grupy mające wpływ na strategię działania firmy.

Powyższe czynniki są ze sobą ściśle powiązane i zmiana w zakresie któregośkolwiek z nich powoduje zmiany w pozostałych. Zmiany stają się w ten sposób przewidywalne co pozwala na rozważenie m. in. negatywnych skutków zmian. Tak dla przykładu można przewidzieć skutki wdrożenia systemu informatycznego (technologia) na kombinację rynek/ produkt, organizację, kadry oraz dominującą koalicję. Z drugiej strony, określenie potrzeby zmian w zakresie produkowanych wyrobów i rynku docelowego identyfikuje systemy informatyczne wspomagające te zmiany.

Również w popularyzacji metod i technik a n a l i z y i p r o j e k t o w a n i a w Polsce istnieje duża dysproporcja w stosunku do bogatego dorobku światowego w tej dziedzinie. Tym niemniej pierwsze opracowania w tym względzie zostały opublikowane [1], [4], [5]. Dlatego w niniejszej pracy nie podjęto szerszej ich charakterystyki. Ogólnie można je

podzielić na modelujące dane i procesy. Typowym przykładem modelowania danych są modele Związków Encji a procesów - diagramy przepływu danych. Najczęściej metody te są jednak kompromisem modeli danych i procesów. Fizycznie mają one postać diagramów lub macierzy.

Pośród około setki tego typu metod i technik największe uznanie zdobyły:

- diagramy Związków Encji (Obiekt-Atrybut-Związek),
- diagramy przepływu danych,
- A-grafy, O-grafy, I-grafy (podejście ISAC),
- diagramy dekompozycji funkcjonalnej,
- modelowanie danych metodą normalizacji,
- diagramy Jacksona,
- diagramy struktury,
- diagramy Nassi-Shneidermana,
- tablice i drzewa decyzyjne,
- diagramy Warniera-Orra.

Listę tę można wydłużyć. Obszerniejsze zestawienie metod, z bardziej dokładną charakterystyką zawierają m.in. opracowania [2], [3], [4].

Niektóre z wyżej wymienionych metod użyteczne są w fazach w d r a ż a n i a i u ż y t k o w a n i a. Należy wymienić tu przede wszystkim diagramy Jacksona i Nassi-Shneidermana w związku z programowaniem oraz prototypowanie i normalizację w odniesieniu do schematów relacyjnych baz danych. W tym względzie znaczącą rolę odgrywają narzędzia czwartej generacji oraz pakiety CASE. Składnikami tych narzędzi są generatory kodów oraz generatory schematów relacyjnych baz danych. Użyteczną techniką stosowaną w tych fazach jest Reverse Engineering, pozwalająca na generowanie dokumentacji analitycznej i projektowej na podstawie kodu źródłowego systemu.



### 3. WARSZTATY METOD I TECHNIK ANALIZY I PROJEKTOWANIA SYSTEMÓW.

Referat niniejszy jest w zasadzie wprowadzeniem w tematykę jednodniowych warsztatów z zakresu podstawowych metod analizy i projektowania systemów. Przebieg warsztatów opiera się na analizie przypadku przedstawionego w Załączniku. Wykonanie poleceń umożliwiających rozwiązanie przypadku będzie wspomagane wykładami, dyskusjami, pracą w grupach oraz odpowiednimi ćwiczeniami. Celem tych warsztatów jest opanowanie przez użytkowników, analityków i projektantów systemów metod i technik, umożliwiających definiowanie w sposób jednoznaczny potrzeb informatycznych a następnie realizację projektów.

Szczegółowy przebieg warsztatów jest następujący:

1. Cel warsztatów i spodziewane wyniki.
2. Tworzenie systemów informatycznych - wykład.
3. Prezentacja i wyjaśnienie wiodącego przypadku.
4. Diagramy przepływu danych (DPD) - wykład.
5. Ćwiczenia z zakresu DPD.
6. Opracowanie diagramu kontekstowego przypadku - praca w grupach.
7. Prezentacja i weryfikacja diagramów kontekstowych przypadku. Dyskusja.
8. Opracowanie diagramu zerowego przypadku - praca w grupach.
9. Prezentacja i weryfikacja diagramów zerowych przypadków. Dyskusja.
10. Modelowanie Związków Encji (obiekt-atrybut-związek) - wykład.
11. Ćwiczenia z zakresu modelowania Związków Encji.
12. Opracowanie diagramu związków encji przypadku - praca w grupach.
13. Prezentacja i weryfikacja diagramów Związków Encji przypadku. Dyskusja.
14. Projektowanie słownika/skorowidza danych - wykład.
15. Opracowanie słownika/skorowidza danych przypadku.
16. Podsumowanie warsztatów.

W trakcie realizacji warsztatów na bieżąco, stosownie do harmonogramu, wręczana będzie jej uczestnikom obszerna dokumentacja, dotycząca treści wykładów i ćwiczeń oraz obejmująca poprawne rozwiązania.

#### BIBLIOGRAFIA.

1. Fuglewicz P., Systemy CASE - problemy, techniki, rozwiązania, w: Komputerowe Wspomaganie Tworzenia Systemów Informatycznych, Trzecia Wiosenna Szkoła PTI, Świnoujście 1990.
2. Martin J., McClure C., Structured Techniques. The Basis for CASE, Prentice Hall, New York 1988.
3. Turner W.S., Langerhorst R.P., Hice G.F., Eilers H.B., Uijtlenbroek, System Development Methodology, Pandata, North-Holland, 1988.
4. Wrycza S., Współczesne metodyki tworzenia systemów informatycznych zarządzania, PTC, Gdańsk 1989.
5. Wrycza S., Aktualne trendy komputerowo wspomaganego tworzenia systemów informatycznych, w: Komputerowe Wspomaganie Tworzenia Systemów Informatycznych, Trzecia Wiosenna Szkoła PTI, Świnoujście 1990.
6. Wrycza S., The Impact of CASE Tools on Teamwork of Information Systems Developers, in Finkelstein A. et al., Human Factors in Analysis and Design of Information Systems, North Holland, Amsterdam 1990, pp. 105-122.
7. Wrycza S., The ISAC-Driven Transition between Requirements Analysis and ER Conceptual Modelling, Information Systems, vol. 15, no.6 (1990), pp. 803-814.



## ZAŁĄCZNIK

### Opis przypadku "Stara Księgarnia"

Celem niniejszego ćwiczenia jest zaprojektowanie systemu informatycznego i skomputeryzowanie "Starej Księgarni".

Firma księgarska "Stara Księgarnia" istnieje od 1925 i specjalizuje się w dostarczaniu książek na zamówienie. W chwili obecnej firma ponownie weszła w prywatne ręce państwa Książkiewiczów.

Pan Książkiewicz zamierza rozwinąć i powiększyć swoje przedsiębiorstwo. Ponieważ ilość zamówień rośnie, firma przestaje być tak efektywna jak kiedyś. Obecnie administrowanie przedsiębiorstwem nie jest wspomagane komputerem. W związku z tym, iż dotychczasowy system zarządzania firmą jest dobry, Pan Książkiewicz zdecydował aby został on skomputeryzowany w całości, bez jakiegokolwiek modyfikacji. Jedynym warunkiem usprawnienia administracji jest to, aby przy tej okazji nie ucierpieli klienci firmy.

Poproszono pana Eksperckiego aby ten pomógł usprawnić "Stara Księgarnię". Pan Ekspercki prześledził działanie przedsiębiorstwa. Z jego raportu wynika, że firma działa w następujący sposób:

"Stara Księgarnia" rozpoczyna swój dzień pracy od zebrania zamówień. Zamówienia te pochodzą od klientów firmy.

Pan Klientowicz sprawdza wszystkie zamówienia klientów, a następnie dla każdego wystawia kartę zamówienia. Na karcie zamówienia notowane są wszystkie dane niezbędne do realizacji zamówienia. Dane o książkach (tj. numer ISBN, wydawca, cena sprzedaży itp.) dostępne są w spisie książek. Spis książek

jest bieżąco uaktualniany o informacje dostarczone przez poszczególnych Wydawców.

Po rejestracji karty zamówień wkładane są do pudełka z przychodzącymi zamówieniami. Na koniec dnia zamówienia te są zabierane przez pana Sprzedawczyńskiego. Dla każdej zamówionej książki pan Sprzedawczyński wystawia kartę zakupu, którą umieszcza w pudełku książek na zakupy.

W końcu tygodnia zawartość tego pudełka jest poddawana do wystawienia zamówień do poszczególnych Wydawców. Aby wystawić takie zamówienie pan Sprzedawczyński potrzebuje informacji o Wydawcy, która to informacja przechowywana jest w odpowiednim katalogu w jego dziale. Karty zakupu, które odpowiadają wystawionym zamówieniom do Wydawców, razem z datą zamówienia, umieszczane są w pudełku z wystawionymi zamówieniami gdzie oczekują na realizację. Książki mogą być dostarczone każdego dnia. Kiedy książki nadchodzą pan Sprzedawczyński używając listu przewozowego, który jest związany z daną przesyłką, przegląda zamówienia do Wydawcy aby sprawdzić jej zawartość. Jeśli przesyłka odpowiada zamówieniom i książka jest w dobrym stanie jest ona przekazywana, wraz z odpowiednią kartą zakupu panu Klientowiczowi. Pan Klientowicz na podstawie kart zamówień wystawia fakturę dla klienta i razem z książką wysyła ją do klienta. Kopia faktury zostaje zarejestrowana i umieszczona w pudełku z niepłaconymi fakturami u pana Pieniążka. Kiedy poczta przynosi potwierdzenia zapłaty, faktura wyjmowana jest z pudełka. Jeśli faktura nie jest zapłacona w ustalonym terminie do klienta wysyła się ponaglenie. W regularnych odstępach czasu pan Pieniążek otrzymuje zastawienie niezapłaconych faktur Wydawców. Faktury te płacone są po sprawdzeniu, że dostarczone książki są w dobrym stanie.



**Polecenie 1.**

Opisz przy pomocy Diagramu Kontekstowego bieżący system informacyjny "Starej Księgarni".

**Polecenie 2.**

Przedstaw bardziej precyzyjnie przy pomocy Diagramu Zerowego system informacyjny "Starej Księgarni".

**Polecenie 3.**

Wykonaj diagram Związków Encji "Starej Księgarni".

**Polecenie 4.**

Podaj pełne i jednoznaczne definicje encji KSIĄZKA i KLIENT.

**Polecenie 5.**

Przy użyciu specyfikacji Potrzeb Informacyjnych stwórz Słownik Dany przypadku "Starej Księgarni".

B. Fornalik, D. Puternicki, W. Staniszkis  
ZETO-RODAN  
Czerniakowska 73/79  
00-718 WARSZAWA

## INŻYNIERIA SYSTEMÓW INFORMATYCZNYCH OD STRATEGII DO REALIZACJI

### 1. WSTĘP

Prezentowano zarys metodyki projektowania i realizacji systemów informatycznych stosowanej i rozpowszechnianej przez ZETO-RODAN. Poza opisem metodyki przedstawiono podstawowe systemy oprogramowania narzędziowego (ProKit\*Workbench, SZBD Ingres) wykorzystywane w pracach projektowo-programowych oraz rozpowszechniane w Polsce przez ZETO-RODAN.

Prezentowana metodyka stanowi ewolucję praktycznych doświadczeń w zakresie projektowania systemów użytkowych opartych o wspólną bazę danych wykorzystywaną przez zespół autorski SZBD RODAN w latach osiemdziesiątych [Roda85] oraz prac badawczych dotyczących metod i technik projektowania systemów informatycznych [Alba85, Ceri83, Olen84]. Przedstawiona w ogólnym zarysie metodyka jest obecnie rozpowszechniana i praktycznie wykorzystywana przez konsultantów ZETO-RODAN. Narzędziem wspierającym prace projektowe jest system ProKit\*Workbench firmy McDonnell Douglas Information Systems, zaś systemem zarządzania bazą danych stanowiącym platformę realizacyjną jest relacyjny system INGRES.

Schemat cyklu życiowego przedsięwzięcia informatycznego przedstawiono w formie schematu przepływu danych na rysunku 1. Wykorzystywana w naszej metodyce notacja schematów przepływu danych jest zgodna z propozycją Chris Gane i Trish Sarson [Gane77]. Wszystkie schematy przepływu danych przedstawione w tej publikacji zostały wykonane przy użyciu edytora graficznego systemu ProKit\*Workbench.

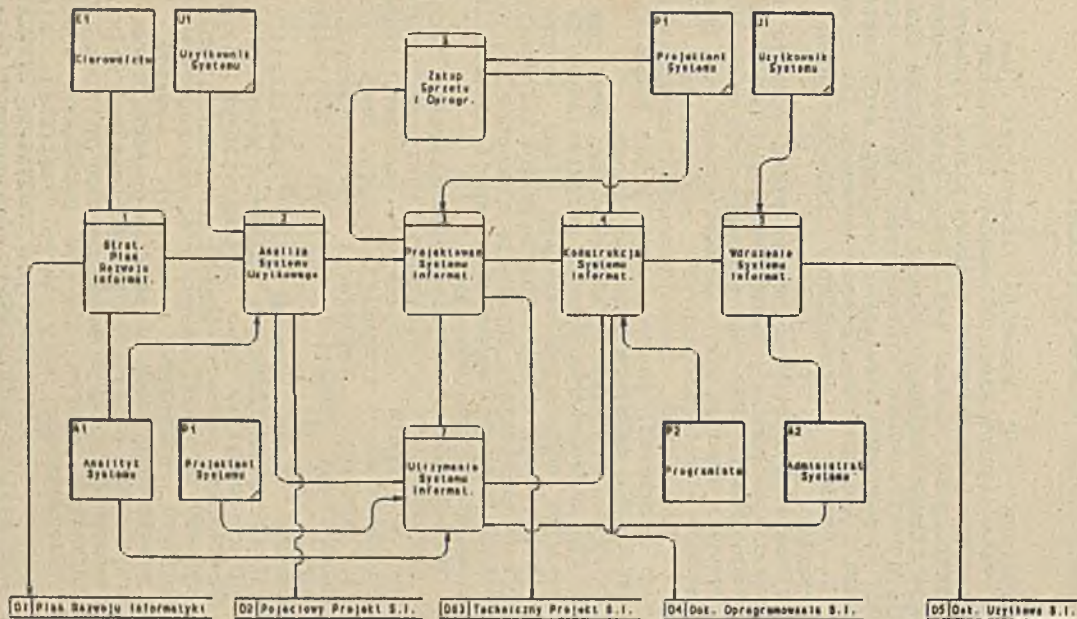
Prezentowany cykl życiowy przedsięwzięcia informatycznego obejmuje wszystkie podstawowe fazy cyklu życiowego systemu informatycznego, zgodnie z analizą wykonaną przez zespół IFIP pod kierownictwem Billa Olle [Olle91], rozpoczynając od strategicznego planu rozwoju informatyki a kończąc na utrzymaniu systemu informatycznego.

W dalszych częściach opracowania przedstawiono trzy podstawowe fazy cyklu życiowego, a mianowicie strategiczny plan rozwoju informatyki oraz analizę i projekt techniczny systemu użytkowego jak również systemy oprogramowania narzędziowego ProKit\*Workbench i SZBD INGRES.

Metodyka ZETO-RODAN jest rozpowszechniana w formie szkoleń i usług wdrożeniowych u użytkowników produktów oprogramowania narzędziowego rozpowszechnianych przez firmę ZETO-RODAN (ProKit\*Workbench, PROIV, INGRES). Licencja pozwalająca na wykorzystanie metodyki jest przyznawana nieodpłatnie wszystkim użytkownikom produktów programowych rozpowszechnianych przez naszą firmę.



# Cykł Życiowy Przedsięwzięcia Informatycznego



Rysunek 1. Zakres metodyki projektowania i realizacji systemów informatycznych

## 2. STRATEGICZNY PLAN ROZWOJU INFORMATYKI

Strategiczny Plan Rozwoju Informatyki jest dokumentem decyzyjnym wyznaczającym kierunki rozwoju zastosowań informatyki w okresie przyjętym jako horyzont czasowy planu. Typowym horyzontem czasowym SPRI jest okres 4-5 lat. Istotą dokumentu jest konfrontacja rzeczywistych problemów i ograniczeń funkcjonowania organizacji z możliwościami wynikającymi z bieżącego stanu rozwoju informatyki. Proponowane rozwiązania obejmują architekturę sprzętu komputerowego i oprogramowania podstawowego jak również zakres funkcji i model informacyjny proponowanych systemów informatycznych.

Proces realizacji strategicznego planu rozwoju informatyki jest przedstawiony w formie schematu przepływu danych na rysunku 2. Struktura dokumentacji SPRI jest przedstawiona na rysunku 3.

Podstawowym elementem planu strategicznego jest preferencyjna lista systemów informatycznych stanowiących rozwiązanie istniejących ograniczeń funkcjonowania organizacji. Lista proponowanych rozwiązań, zaakceptowana przez kadrę kierowniczą, stanowi podstawę dla dalszych prac nad planem.

Punktem wyjściowym części informatycznej SPRI jest definicja architektury systemów informatycznych przewidzianych do realizacji w okresie objętym horyzontem czasowym planu. Ważnym elementem prezentacji proponowanej architektury systemów informatycznych jest tablica krzyżowa ilustrująca korelację elementów tej architektury z ograniczeniami funkcjonowania organizacji zidentyfikowanymi w trakcie analizy stanu istniejącego.

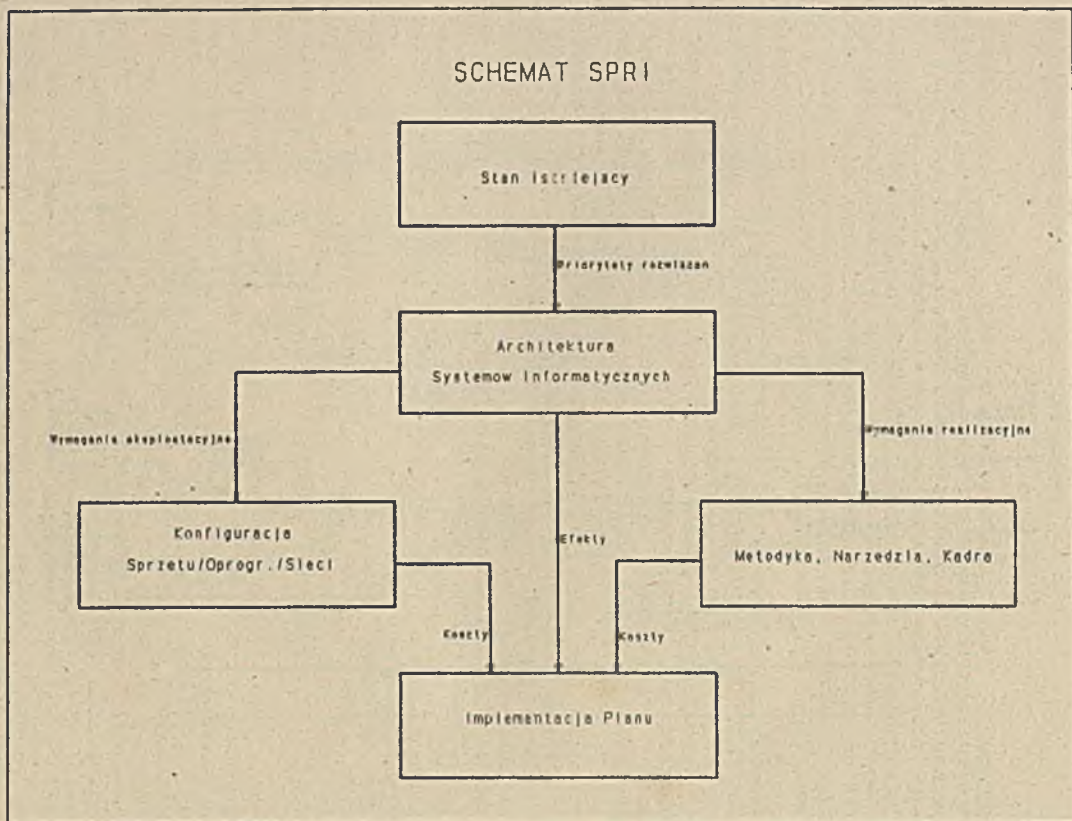
Dalsze części planu obejmują uwarunkowania technologiczne realizacji proponowanej architektury systemów informatycznych przedstawione w formie architektury sprzętu komputerowego, oprogramowania oraz sieci transmisji danych, oraz w formie opisu metodyki i narzędzi realizacji proponowanych systemów informatycznych jak również wymagań dotyczących kadry informatycznej.

Podsumowanie planu stanowi plan realizacji obejmujący podstawowe fazy przewidzianych planem prac nad systemami informatycznymi. Specyfika prac informatycznych nie pozwala na dokładne określenie pracochłonności wszystkich kolejnych faz realizacji systemu informatycznego. W naszej metodyce proponujemy metodę kolejnych przybliżeń estymacji kosztów realizacji przyjmującą zasadę dokładnego określania pracochłonności następnej fazy oraz przybliżone szacowanie pracochłonności dalszych etapów pracy. Ostateczna akceptacja proponowanego strategicznego planu rozwoju informatyki odbywa się na podstawie analizy planu finansowego obejmującego cały okres realizacji planu. Plan finansowy powinien określać wszystkie podstawowe elementy kosztów realizacji oraz analizę opłacalności proponowanych przedsięwzięć.





Rysunek 3. Struktura dokumentu strategicznego planu rozwoju informatyki.





### 3. ANALIZA SYSTEMU UŻYTKOWEGO

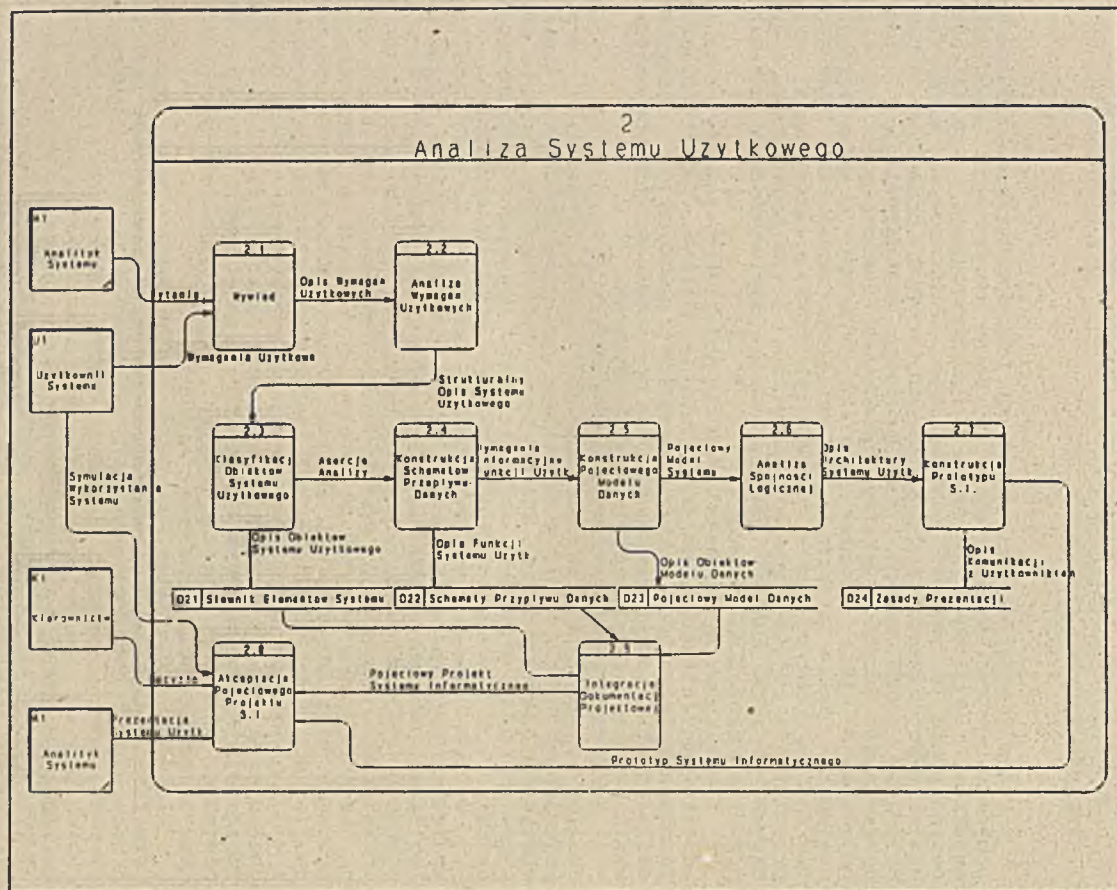
Celem analizy systemu użytkowego jest opracowanie pojęciowego projektu systemu pozwalającego na pełną prezentację wszystkich aspektów użytkowych systemu informatycznego wynikających bezpośrednio z wymagań użytkowników. Podstawowymi elementami projektu pojęciowego są słownik danych, schematy przepływu danych, pojęciowy model danych, oraz prototyp przedstawiający zasady prezentacji danych. Istotnym elementem dokumentacji projektu pojęciowego jest również analiza wymagań eksploatacyjnych obejmująca takie elementy jak definicja ograniczeń czasowych funkcji użytkowych (czas odpowiedzi, tryb przetwarzania) oraz przewidywana zajętość pamięci zewnętrznej przez podstawowe zbiory danych. Proces analizy jest przedstawiony w formie schematu przepływu danych na rysunku 4.

Słownik danych obejmuje wszystkie podstawowe elementy projektu pojęciowego (atrybuty, obiekty, funkcje użytkowe) identyfikowane jednoznacznie przez nazwy elementów słownika i opatrzone definicjami. Słownik danych systemu użytkowego stanowi podstawowy wynik procesu klasyfikacji elementów systemu.

Konstrukcja schematów przepływu danych prezentujących wynik analizy funkcji systemu użytkowego (aspekt dynamiczny) oraz pojęciowego modelu danych stanowiącego wynik analizy danych (aspekt statyczny) jest procesem iteracyjnym. Poszczególne iteracje obu typów analiz prowadzą do kolejnych uściśleń reprezentacji dynamicznych i statycznych aspektów systemu informatycznego.

Ważnym etapem analizy systemu użytkowego jest analiza spójności logicznej pojęciowego modelu systemu. Analiza spójności logicznej jest realizowana automatycznie przez funkcje poziomego i pionowego bilansowania schematów przepływu danych zapisanych w encyklopedii (repository) systemu ProKit\*Workbench. Analiza pozwala na identyfikację wszystkich błędów formalnych w specyfikacji procesów oraz strumieni i zbiorów danych. Dodatkowym elementem analizy spójności logicznej jest konfrontacja asercji wynikających z analizy wymagań użytkowych z opisem pojęciowego modelu danych obejmującym pojęciowy schemat danych oraz definicję reguł spójności logicznej. Dopuszczalne notacje graficzne pojęciowego modelu danych obejmują schematy struktury danych, schematy obiekt-powiązanie Chen'a oraz pojęciowe schematy relacyjne.

Istotnym elementem projektu pojęciowego jest prototyp interface'u użytkownika pozwalający na pełną prezentację ekranów proponowanego systemu, nawigacji po drzewie funkcji użytkowych systemu, oraz diagnostyki i funkcji help. Konstrukcja prototypu w ramach projektu pojęciowego jest podstawowym warunkiem uzyskania świadomej akceptacji użytkownika co z kolei prowadzi do znacznego uproszczenia fazy wdrażania systemu użytkowego. Prototyp jest realizowany całkowicie przy wykorzystaniu odpowiednich funkcji systemu ProKit\*Workbench.



#### Rysunek 4. Proces analizy systemu użytkowego



#### 4: PROJEKT TECHNICZNY SYSTEMU UŻYTKOWEGO

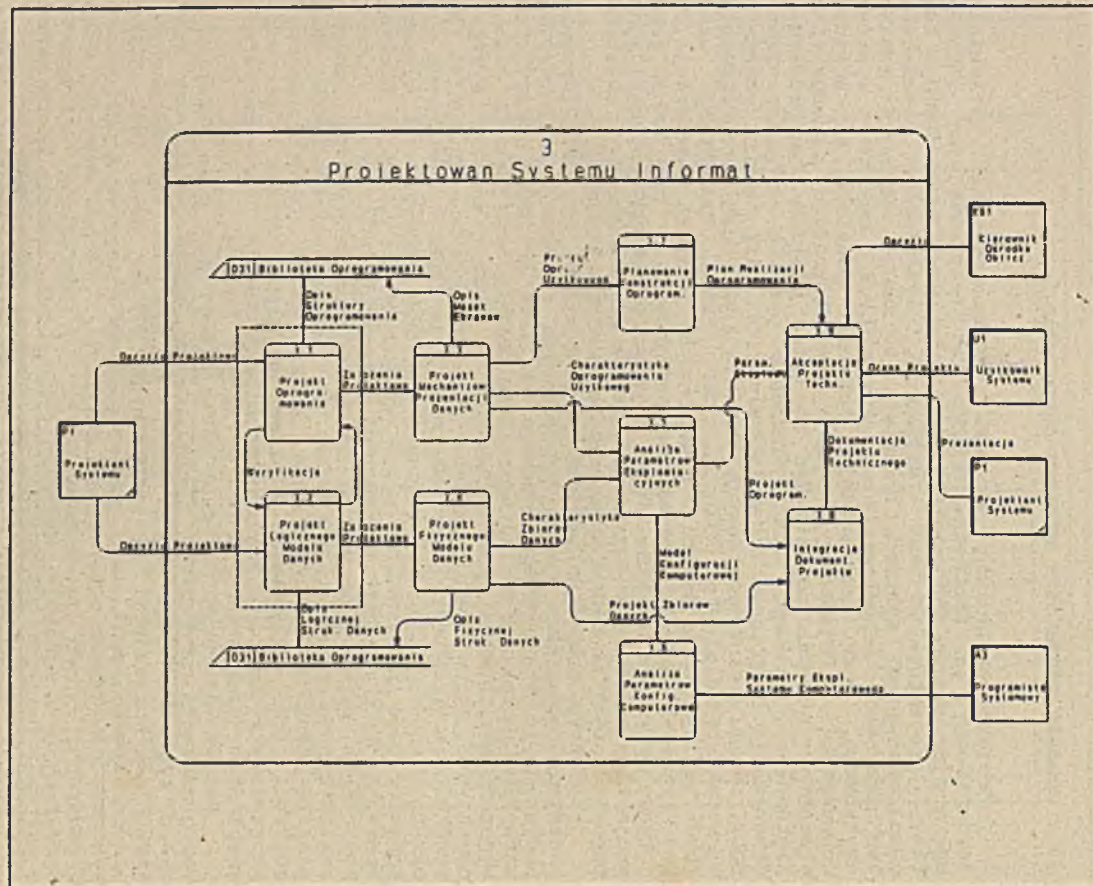
Projekt techniczny systemu użytkowego jest całkowicie zorientowany na konkretne środowisko eksploatacyjne (system operacyjny, system ZBD, monitor TP) a podstawowym ograniczeniem tej fazy projektowej jest uzyskanie wymaganych parametrów eksploatacyjnych systemu. Przyjęta w naszej metodycie zasada jednokrotnego definiowania poszczególnych aspektów systemu użytkowego, przy zastosowaniu ścisłego podziału na poziomy dokumentacji projektowej, wymaga automatycznego przenoszenia elementów projektu pojęciowego do opisu projektu technicznego. Zastosowanie systemu ProKit\*Workbench pozwala na automatyczne generowanie logicznego schematu relacyjnej bazy danych (standard SQL), opisów ekranów użytkowych, definicji struktury oprogramowania i struktury dialogów (PRO IV, CSP, TELON, TRANSFORM), oraz deklaracji struktur danych w podstawowych językach 3 generacji. W tej sytuacji decyzje projektowe podejmowane w tej fazie prac dotyczą wyłącznie aspektów determinujących parametry eksploatacyjne projektowanego system. Zakres parametrów eksploatacyjnych objętych typowym projektem technicznym przedstawiono w [Stan84].

Istotnym elementem projektu technicznego jest przewidywanie parameterów eksploatacyjnych wynikających z decyzji projektowych podjętych w ramach rozpatrywanego wariantu projektu. Oczywistym warunkiem akceptacji wariantu projektu technicznego jest spełnienie wymagań eksploatacyjnych ustalonych w ramach analizy systemu użytkowego. Przewidywanie wartości parametrów czasowych, takich jak średni czas odpowiedzi poszczególnych klas funkcji użytkowych czy przepustowość systemu wymaga oceny obciążeń konfiguracji systemu komputerowego generowanych przez oprogramowanie użytkowe (liczba operacji I/O, liczba instrukcji jednostki centralnej). Ocena powyższych parameterów wymaga realizacji dwupoziomowego modelu analitycznego obejmującego analizę obciążeń (równania kosztów) oraz analizę funkcjonowania systemu komputerowego (modele masowej obsługi). Opis stosowanych w naszej metodycie technik i narzędzi programowych obejmujących hierarchiczne modele oceny paramtrów eksploatacyjnych przedstawiono w [Alba85, Orla89].

Ważnym elementem projektu technicznego jest plan realizacji oprogramowania użytkowego obejmujący wszystkie podstawowe fazy cyklu życiowego tego etapu prac nad systemem informatycznym. Warunkiem konstrukcji realistycznych planów prac jest ocena pracochłonności realizacji poszczególnych elementów architektury systemu. Wytyczne dotyczące estymacji pracochłonności oraz organizacji prac programowych są przedstawione w sposób wyczerpujący w [Ghez91]. Wszystkie podstawowe elementy proponowane w tej pracy zostały wprowadzone do metodyki ZETO-RODAN.

Proces akceptacji projektu technicznego obejmuje takie elementy jak ocena parameterów eksploatacyjnych systemu użytkowego, ocena planu realizacji oprogramowania, oraz ustalenie zasad kontroli jakości oprogramowania.

Rysunek 5. Proces projektowania systemu użytkowego





## 5. ProKit<sup>®</sup>WORKBENCH<sup>®</sup>

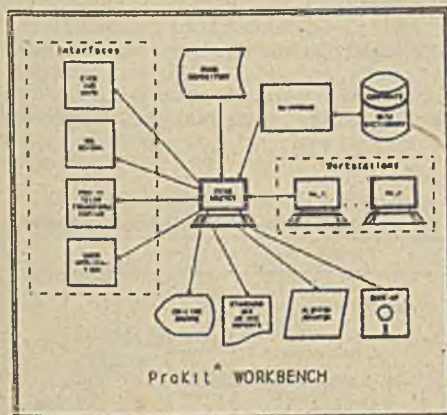
System ProKit<sup>®</sup>WORKBENCH<sup>®</sup> jest zaawansowanym, zintegrowanym narzędziem typu CASE (Computer Aided System Engeneering) wspomagającym analityka i projektanta w trakcie całego cyklu życiowego systemu informatycznego cd planowania strategicznego poprzez analizę i projektowanie aż do utrzymania.

Architektura systemu oparta jest o koncepcję stacji roboczej projektanta. Podstawowym elementem systemu jest obiektowo zorientowana baza danych (encyklopedia) zawierająca pełną dokumentację projektową. Baza ta może być utrzymywana w ramach jednej stacji roboczej lub może być wspólna dla całego zespołu projektowego, dostępną w ramach sieci lokalnej.

System ProKit<sup>®</sup>WORKBENCH<sup>®</sup> działa na komputerach klasy IBM PC 286, 386 z pamięcią rozszerzoną do 2 MB w środowisku MS/PC-DOS lub w sieciach: Novell, OS/2 LAN, Banyan-Vinece.

Podstawowe funkcje systemu oparte są o metodykę analizy systemów Gane & Sarson i obejmują analizę funkcji i danych systemu, projekt pojęciowy bazy danych, projekt interface'u użytkownika oraz projekt struktury oprogramowania. Wszystkie informacje projektowe są wprowadzane we wpelni zintegrowanym środowisku tekstowo-graficznym.

Zakres funkcji projektowanego systemu reprezentowany jest graficznie przy pomocy wielopoziomowych schematów przepływu danych (Data Flow Diagrams), a projekt pojęciowy bazy danych przy wykorzystaniu standardowej notacji Entity/Relationship (Chena) lub w postaci schematów Bachmana. Interface użytkownika jest prezentowany dynamicznie poprzez funkcje szybkiego prototypowania i symulacji jego działania dając już na etapie projektu systemu możliwość weryfikacji funkcji konwersacyjnych. Projekt struktury oprogramowania reprezentowany jest graficznie przy pomocy wielopoziomowych sche-



matów strukturalnych (Structure Charts).

System posiada również interface'y do różnych środowisk systemowo-programowych. Projekt logiczny relacyjnej bazy danych jest generowany automatycznie w oparciu o zasady odwzorowania pojęciowego modelu danych w strukturę relacyjną w standardzie ISO/SQL (dla Ingres, DB2, Oracle, Informix). Opisy struktur ekranów są również generowane automatycznie zarówno jako deklaracje danych w językach programowania (Pascal, C, Cobol, Ada, Fortran, Basic, Assembler) jak i definicje map ekranów dla systemów CICS, IMS/PC.

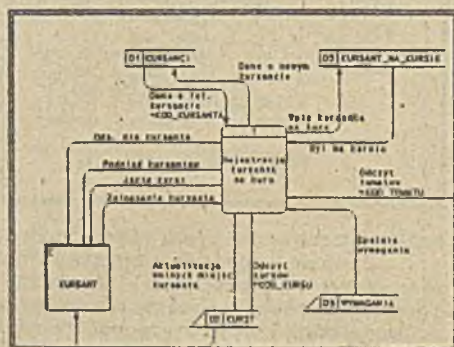
Faza konstrukcji oprogramowania użytkowego wspomagana jest poprzez interface'y do różnych generatorów aplikacji (PRO-IV®, CSP/AD®, Telon®, Transform®), które to przenoszą do tych środowisk większość informacji projektowych. System posiada również bardzo bogate narzędzia do tworzenia raportów (standardowe, Ad/hoc) wyprowadzanych na drukarkę lub do zbioru i dokumentacji graficznej rysowanej na ploterze.

### Podstawowe cechy systemu ProKit\*WORKBENCH®

- \* obiektowo identyfikowana encyklopedia systemu,
- \* bogate środowisko graficzne zintegrowane z encyklopedią systemu:
  - edytor schematów przepływu danych,
  - edytor pojęciowego modelu danych,
  - edytor schematów strukturalnych,
  - edytor diagramów prezentacyjnych,
- \* prototypowanie i symulacja interface'u użytkownika,
- \* bogata dokumentacja projektowa:
  - raporty standardowe,
  - raporty Ad/hoc,
  - dokumentacja graficzna,
- \* funkcje IMPORT/EXPORT,
- \* interface'y do różnych generatorów aplikacji (PRO-IV®, CSP/AD®, Telon®, Transform®),
- \* dostępna wersja sieciowa (Multi-user),
- \* możliwa praca wielostanowiskowa (Multi-workstation),
- \* obsługa bardzo dużych projektów (encyklopedia systemu do 128 MB),
- \* wielowersyjność projektu,
- \* pomoc 'on-line',
- \* obsługa różnych monitorów (mono,color), drukarek (igłowe, laserowe), ploterów (A0-A4).

### Cechy modułu ANALIZATOR'a

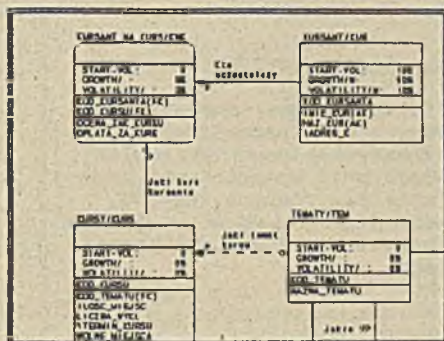
- \* 11 poziomów schematów przepływu danych (Gane & Sarson),
- \* bilansowanie poziome i pionowe schematów,
- \* synchronizacja pomiędzy modelem funkcji a modelem danych,
- \* max. 1600 symboli na jednym poziomie schematu.





## Cechy modułu DATA MODELER'a

- \* schematy obiekt-powiązanie (rozszerzony Chen lub Bachman),
- \* schematy relacyjne (rozszerzone IDEF1X),
- \* wspomaganie normalizacji z migracją kluczy,
- \* generowanie schematu logicznego relacyjnej bazy danych,
- \* synchronizacja między modelem danych a modelem funkcji,
- \* max. 1600 obiektów na schemacie modelu danych,
- \* możliwy pełny opis obiektów na schemacie (klucze, atrybuty, charakterystyka).

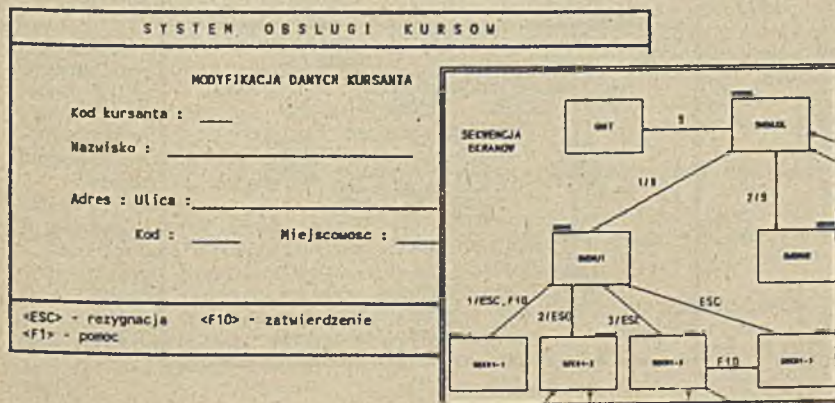


## Cechy modułu PROTOTYPER'a

- \* projektowanie ekranów,
- \* symulacja interface'u użytkownika poprzez tworzenie sekwencji ekranów, warunków poruszania się po nich i sposobu ich wypełniania,
- \* generowanie wersji źródłowych definicji danych w jednym z 7 języków (Pascal, C, Cobol, Ada, Fortran, Basic, Assembler),
- \* generowanie map ekranów dla CICS, IMS/DC,
- \* synchronizacja między ekranami prototypu a modelem funkcji,
- \* pełna dokumentacja graficzna i tekstowa zaprojektowanych ekranów.

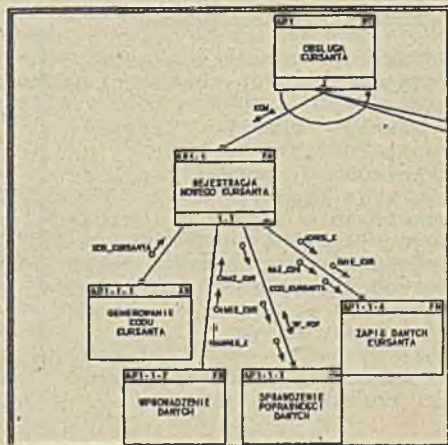
-- IMAGE HARDCOPY PROLOGUE % CREATED BY ProKit\*WORKBENCH

/EK1-3/



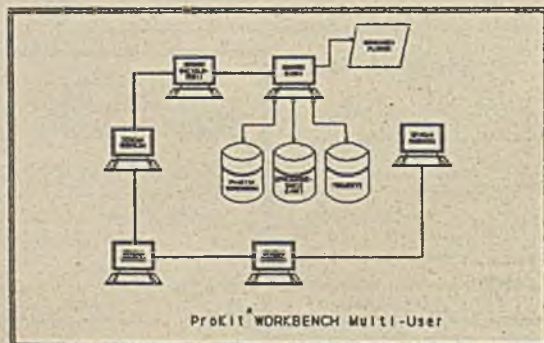
Cechy modulu DESIGNER'a

- \* wielopoziomowe schematy strukturalne (Constantine),
- \* wykorzystywanie wyników analizy,
- \* pomoc przy specyfikacji programów dla różnych języków,
- \* max. 1600 obiektów na jednym poziomie schematu strukturalnego.



Cechy wersji sieciowej ProKit® WORKBENCH®

- \* architektura klient/serwer,
- \* równoległa obsługa wielu projektów i wersji w jednej sieci,
- \* synchronizacja pomiędzy encyklopedią a diagramami w czasie rzeczywistym,
- \* blokowanie na poziomie obiektów, a nie rekordów,
- \* log transakcji,
- \* 'page image posting',
- \* możliwość szybkiego 'recovery',
- \* równoważność funkcji użytkowych z wersją jedno-użytkownikową,
- \* wszystkie cechy wersji jedno-użytkownikowej.





Zastosowanie systemu ProKit\*WORKBENCH® podnosi w sposób znaczący jakość realizowanych systemów informatycznych oraz znacznie obniża koszty opracowania. Obniżenie kosztów i czasu realizacji wynika z wprowadzenia sformalizowanej metodyki projektowania wspomaganej przez poszczególne funkcje systemu, usprawnienia komunikacji między projektantami a użytkownikiem jak również w obrębie zespołu projektowego, oraz możliwości łatwej prezentacji projektu użytkownikowi końcowemu. Kompletność dokumentacji projektowej oraz automatyczna kontrola spójności logicznej poszczególnych elementów projektu pozwala na wykrywanie błędów już we wczesnych fazach realizacji systemu informatycznego. Dzięki takiemu narzędziu usprawnieniu ulega proces dokumentowania prac projektowych a tym samym utrzymanie i koserwacja systemu stają się prostsze.

System ProKit\*WORKBENCH® znalazł już zastosowania w wielu różnych jednostkach gospodarczych takich jak przedsiębiorstwa, instytuty naukowo-badawcze, uczelnie, banki, biura projektów zarówno w kraju jak i za granicą. Jednocześnie należy podkreślić, że firma McDonnell Douglas Information Systems, właściciel i twórca systemu ProKit\*WORKBENCH®, systematycznie rozwija i doskonali swój produkt tym samym dając nam szansę stosowania narzędzia, które zawsze znajduje się w światowej czołówce CASE'ów.

## 6. SZBD Ingres

### 6.1. Wprowadzenie

Ingres albo Ingres Intelligent Database System jest, z punktu widzenia stanu sztuki, niekwestionowanym liderem wśród zaawansowanych systemów zarządzania relacyjną bazą danych (relational database management system).

Aktualnie są oferowane cztery wersje Ingresu: 5.0, 6.2, 6.3 i 6.4. Pierwsze dwie wersje (5.0 i 6.2) zawierają podstawowy system zarządzania relacyjną bazą danych (RDBMS), który zawiera dwa relacyjne języki zapytań SQL i QUEL. Natomiast dwie ostatnie wersje (6.3 i 6.4) zawierają dwa dodatkowe rozszerzenia: zarządzanie wiedzą (Knowledge Management) i zarządzanie obiektami (Object Management). Te trzy produkty: RDBMS, Knowledge Management i Object Management są sprzedawane oddzielnie.

Produkty Ingresu, ze względu na przeznaczenie, dzieli się na cztery grupy:

- \* zintegrowany zbiór narzędzi wspomagających decyzje,
- \* zintegrowany zbiór narzędzi wspomagających budowę oprogramowania,
- \* produkty komunikacji,
- \* produkty dla komputerów osobistych.

Ingres może być eksploatowany na ponad 40 platformach, w tym pod następującymi systemami operacyjnymi: Micro VMS 4.7 i 5.X, VMS 4.7 i 5.X, DOS, OS/2, UNIX, AIX, SunOs, HP-UX, DEC ULTRIX, SCO UNIX.

## 6.2. Zarządzanie danymi (server)

Podstawowy system zarządzania relacyjną bazą danych (RDBMS) obsługuje relacyjną bazę danych. Baza danych z punktu widzenia użytkowników składa się z tablic dwuwymiarowych (relacji), a tablice z wierszy (krotek, n-tek) i kolumn (atrybutów). Tablica może zawierać, w najnowszej wersji, do 300 kolumn. Liczba wierszy w tablicy jest nieograniczona.

Dopuszczalne są następujące typy danych:

- \* dane numeryczne stałoprzecinkowe (integer1, integer2, integer4) i zmiennoprzecinkowe (float4, float8),
- \* dane alfanumeryczne stałej długości (char) oraz zmiennej długości (varchar),
- \* dane abstrakcyjne: date i money.

Na poziomie bazy danych można zdefiniować reguły integralności danych, które umożliwiają utrzymanie poprawnych danych w bazie danych. Istnieje możliwość tworzenia procedur bazy danych, które z kolei zwiększają wydajność systemu.

Jednostką przesyłania danych pomiędzy pamięcią operacyjną a bazą danych (pamięcią dyskową) jest 2 Kb stronica.

Tablica (struktura logiczna) może być zapisana na dysku jako jedna z czterech następujących struktur fizycznych:

- \* heap (zbiór sekwencyjny o pewnych specyficznych własnościach),
- \* hash (zbiór haszowany, zbiór o organizacji wymieszanej);
- \* ISAM (zbiór indeksowy (statyczny)),
- \* B-tree (zbiór indeksowy (dynamiczny)).

Struktury fizyczne będące odwzorowaniem tablic w pamięci dyskowej mogą być dowolnie zmieniane. Dla zbiorów danych o organizacji ISAM i heap mogą być tworzone indeksy wtórne.

Duże bazy danych mogą być rozmieszczane na wielu jednostkach pamięci dyskowej.

Ingres wykorzystuje optymalizator zapytań (query optimizer) do doskonalenia strategii realizacji zapytań, a w konsekwencji tego do wzrostu wydajności systemu.

Do zabezpieczenia systemu przed fizycznym zniszczeniem wykorzystuje się między innymi programowe zabezpieczenia polegające na:

- \* tworzeniu kopii zapasowych systemu i baz danych (backups),
- \* tworzeniu statycznych punktów kontrolnych bazy danych (checkpoints),
- \* prowadzeniu dzienników zmian dokonanych w bazie danych (journals).

Poprawność współbieżnego wykonywania transakcji zapewnia system blokad. Stosuje się dwa rodzaje blokad:

- \* blokowanie dzielone (shared lock) dla odczytu,
- \* blokowanie wybiórcze (exclusive lock) dla zapisu.

Blokowanie może być realizowane na trzech poziomach:

- \* tablicy (table level),
- \* stronicy (page level),
- \* rekordu - wiersza tablicy (record level).

Dostęp do danych w bazie danych jest określany na trzech poziomach:



- \* administratora systemu Ingres,
- \* administratora bazy danych,
- \* użytkownika Ingresu.

Administrator systemu Ingres otrzymuje zwykle uprawnienia super użytkownika oraz dostęp do wszystkich baz danych i tablic w bazach danych. Administratorem bazy danych staje się każdy użytkownik Ingresu, który utworzył bazę danych. Ma on dostęp do wszystkich własnych baz danych. Ponadto określa on uprawnienia pozostałych użytkowników Ingresu w zakresie dostępu do swoich baz danych. Użytkownik Ingresu ma dostęp do wszystkich baz danych określonych jako publiczne (public) oraz do baz danych i operacji na danych przydzielonych przez poszczególnych administratorów baz danych.

Słownik danych jest automatycznie tworzony i utrzymywany przez Ingres. Stanowi go system katalogów i tablic, które są aktywne podczas eksploatacji bazy danych. W tablicach tych są zapisywane między innymi dane o bazach danych, tablicach w bazach danych, atrybutach, użytkownikach i ich przywilejach, reguły integralności, dane o indeksach, dane dla optymalizatora zapytań. Dane ze słownika danych są dostępne dla użytkowników Ingresu za pomocą komendy select języka SQL lub komendy retrieve języka QUEL.

### 6.3. Zarządzanie wiedzą

System zarządzania wiedzą (Knowledge Management) umożliwia utworzenie bazy wiedzy, która może zawierać:

- \* system reguł,
- \* system sterowania zasobami systemu,
- \* system sterowania dostępem do baz danych.

System reguł powinien zawierać reguły działalności ("business rules") obowiązujące w dziedzinach przedmiotowych odwzorowanych w bazy danych. Mogą one być związane z jedną lub wieloma tablicami. Reguły są opisywane w procedurach, które są wykonywane w razie wystąpienia pewnych zdarzeń w bazie danych. Najczęściej są to zdarzenia związane z wykonaniem operacji wprowadzenia (insert), modyfikowania (modify) i usunięcia (remove) danych.

System sterowania zasobami umożliwia administratorowi bazy danych dysponowanie zasobami systemu podczas wykonywania zapytań (queries). Może on określić dla danego użytkownika:

- \* limit kosztu,
- \* limit czasu CPU,
- \* liczbę dostępnych stronic,
- \* liczbę blokad,
- \* liczbę dostępnych wierszy w tablicy.

System sterowania dostępem do baz danych umożliwia określenie dostępu wybranym użytkownikom lub grupom użytkowników do:

- \* tablic w zakresie wskazanych operacji w odniesieniu do wybranych kolumn lub wierszy,
- \* tablic wirtualnych (views),
- \* programów (role permission),
- \* baz danych.

#### 6.4. Zarządzanie obiektami

System zarządzania obiektami (Object Management) umożliwia użytkownikowi Ingresu zdefiniowanie:

- \* własnych typów danych (nieдоступnych dotychczas w Ingresie),
- \* dodatkowych funkcji,
- \* nowych operatorów.

#### 6.5. Narzędzia wspomagające decyzje

Ingres oferuje następujący zintegrowany zestaw narzędzi wspomagających decyzje:

- \* Ingres/Menu,
- \* Query By Forms (QBF),
- \* Report By Forms (RBF),
- \* Graph By Forms (GBF),
- \* Ingres/Window View,
- \* Graphical Query Language.

Ingres/Menu jest interfejsem do narzędzi Ingresu. Umożliwia on użytkownikowi pełny dostęp do bazy danych za pomocą kolejnych ekranów zawierających listy komend (menu). Komendy te umożliwiają między innymi manipulowanie danymi (retrieve, add, delete, modify), manipulowanie tablicami, tworzenie raportów, projektowanie formularzy, tworzenie grafiki i programów.

Query By Forms umożliwia użytkownikowi za pomocą kolejnych ekranów dostęp do bazy danych w celu wyszukiwania lub aktualizowania danych. Podstawową cechą tego narzędzia jest to, że użytkownik nie poluguje się komendami relacyjnego języka zapytań ale edytorem graficznym. Zapytania są realizowane dwufazowo. W fazie pierwszej użytkownik definiuje zapytanie (Join Definition), zaś w fazie drugiej wykonuje je (Query Execution). Faza druga może być wykonywana wielokrotnie w dowolnym momencie czasowym.

Report By Forms służy do generowania raportów w układzie kolumnowym i blokowym.

Graph By Forms służy do przedstawiania danych w postaci graficznej.

Ingres/Window View umożliwia wykonanie programów obsługi terminali znakowych w niezmiennionej postaci na stacji roboczej.

Graphical Query Language umożliwia dostęp do danych Ingresu z komputera Macintosh oraz integrację jego oprogramowania z Ingresem.

#### 6.6. Narzędzia wspomagające budowę oprogramowania

Ingres oferuje między innymi następujące narzędzia wspomagające budowę oprogramowania:

- \* Applications By Forms (ABF),
- \* Visual Form Editor (VIFRED),
- \* Report Writer,
- \* Ingres/Windows 4GL,



- \* Ingres/Vision,
- \* Host Language Interfaces.

Applications By Forms służy do budowy programów. ABF program składa się ze zbioru powiązanych ramek (frames) i procedur. Ramki mogą być wygenerowane przez QBF, GBF i RBF. Ramka wizualnie składa się z formularza (form) oraz listy komend (menu) skojarzonej z danym formularzem. Procedury mogą być napisane w języku Ingres/4GL, języku trzeciej generacji lub języku SQL albo w języku QUEL. Za podstawowy jest uważany język Ingres/4GL. Zawiera on cztery kategorie zdań:

- \* zdania sterujące wykonywaniem programu,
- \* zdania sterujące formularzem,
- \* zdania dotyczące bazy danych,
- \* zdania zapytań.

Zdania sterujące wykonywaniem programu są podobne do zdań tego typu w językach trzeciej generacji. Są to między innymi następujące zdania: callframe, callproc, if-then-else, while-do, return, exit.

Zdania sterujące formularzem sterują wykonywaniem ramek. Przykładowo są to następujące zdania: assignment, clear, clearrow, deleterow, helpfile, inittable, insertrow, message, mode, next, printscreen, prompt, resume, set-form, sleep, validate, validrow.

Zdania dotyczące bazy danych zawierają wirtualnie pełny język SQL lub QUEL. Zdania jednego z tych języków są włączane do podprogramów języka Ingres/4GL.

Zdania zapytań są szczególnymi zdaniami zapytań. Udostępniają one dane o przebiegu realizacji programów i zapytań w języku SQL lub QUEL. Są to następujące zdania: inquire\_forms i inquire ingres. Ponadto do tej grupy należą zdania: validrow, insertrow, clearrow, deleterow i repeated select.

Visual Form Editor służy do dostosowania istniejących ramek do potrzeb danego ABF programu. Umożliwia on projektantowi programu zmodyfikowanie układu formularza, zdefiniowanie komunikatu o błędzie dla danego pola, zmienienie nazwy i atrybutów pola.

Report Writer jest językiem projektowania i generowania złożonych raportów. Dane do raportów są dostarczane z bazy danych za pomocą komend języka SQL lub QUEL.

Ingres/Window 4GL dostarcza użytkownikowi graficzny interfejs do budowy oprogramowania.

Ingres/Vision jest narzędziem umożliwiającym użytkownikowi Ingresu tworzenie, ulepszanie i modyfikowanie programów bez używania języka programowania.

Host Language Interface. Relacyjne języki zapytań mogą być zagnieżdżane, w zależności od systemu operacyjnego, w różnych językach programowania trzeciej generacji. Użytkownik może tworzyć dowolne jednostki programowe za pomocą narzędzi Ingresu, a następnie je włączać do programów w bazowym języku 3GL (host language) za pomocą zdania CALL.

## 6.7. Produkty komunikacji

• Ingres oferuje następujące produkty komunikacji:

- \* Ingres/Net,
- \* Ingres/Star,
- \* Gateways,
- \* Ingres/PCLink.

Ingres/Net jest produktem, umieszczanym na poszczególnych węzłach sieci. Pojedynczy produkt umożliwia podłączenie do danego węzła do 100 użytkowników Ingresu. Każdy użytkownik Ingresu może mieć w danym momencie czasu dostęp do jednego lub kilku węzłów sieci, a tym samym do jednej lub kilku baz danych znajdujących się w różnych węzłach. Operuje on między innymi na następujących protokołach: TCP/IP, DECnet, X.25, PC-LANS, NetBIOS.

Ingres/Star jest produktem zarządzającym rozproszoną bazą danych (baza danych znajdująca się w różnych węzłach sieci). Umożliwia on użytkownikowi, w danym momencie czasu, dostęp do baz danych znajdujących się w różnych węzłach sieci. Zawiera on narzędzia umożliwiające eksploatację rozproszonego słownika danych oraz wykonywanie rozproszonych transakcji. Ponadto, jako jedyny w tej chwili system, jest wyposażony w statystyczny optymalizator rozproszonych zapytań, dla których opracowuje scenariusze ich realizacji optymalizując wykorzystanie zasobów systemu.

Gateways jest produktem umożliwiającym użytkownikowi dostęp do nie-Ingres-owych baz danych, np. DB2, SQL/DS, Rdb, Teradata, Tandem, HP Allbase, Data General, IMS, RMS. Językiem dostępu jest OPEN SQL zawierający około 85% możliwości języka podstawowego.

Ingres PCLink umożliwia dostęp do Ingres-owych baz danych za pomocą oprogramowania komputerów osobistych, w tym Lotus'a 1-2-3, WordPerfect'u, pakietu dBase.

## 6.8. Produkty dla komputerów osobistych

Ingres oferuje niektóre swoje produkty w wersji na komputery osobiste. Umożliwia to zintegrowanie masowo eksploatowanych komputerów osobistych ze środowiskiem Ingresu oraz dostęp do baz danych. W ramach tej grupy są dostępne między innymi następujące produkty:

- \* VisionPro,
- \* Embedded SQL dla języka C.

VisionPro jest zintegrowanym pakietem zawierającym dwa produkty uprzednio scharakteryzowane, tj.: INGRES/Vision i INGRES/4GL. Umożliwia on użytkownikowi tworzenie, szybko i tanio, programów.



# BIBLIOGRAFIA

- Alba85 Albano, A., De Antonellis, V., Di Leva, A., (Eds), Computer-Aided Database Design - the DATAID Project, North-Holland 1985.
- Ceri83 Ceri, S. (Ed.), Methodology and Tools for Data Base Design, North-Holland 1983.
- Gane77 Gane, C., Sarsen, T., Structures Systems Analysis: Tools & Techniques, McDonnell Douglas Corporation 1977.
- Ghez91 Ghezzi, C., Jazayeri, M., Mandrioli, D., Fundamentals of Software Engineering, Prentice-Hall International Editions 1991.
- Hamm88 Hammond, J.L., Performance Analysis of Local Computer Networks, Addison Wesley 1988.
- Lazo84 Lazowska E.D., Zahorian, J., Graham, G.S., Sevcik, K.C., Quantitative System Performance - Computer System Analysis Using Queueing Network Models, Prentice-Hall 1984.
- Most90 Mostardi, T., Staniszkis, W., Multidatabase System Design Methodology, Database Technology, Vol. 3, Nr 1.
- Olen84 Olenki, J., Staniszkis, W., Projektowanie bazy danych, PWE 1984.
- Olle91 Olle, T.W., Hagelstein, J., Macdonald, I.G., Rolland, C., Sol, H.G., Van Assche F.J.M., Verrijn-Stuart, A.A., Information Systems Methodologies - A Framework for Understanding, Addison Wesley 1991.
- Roda85 Zespół RODAN, Metodyka Projektowania Wspólnej Bazy Danych, ZETO-ZOWAR 1985.
- Orla89 Orlando, S., Perri, V., Scrivano, S., Staniszkis, W., Database Analyzer and Predictor - An Overview, Proc. Fifth Int. Conf. on Data Engineering, Los Angeles, Luty 1989, IEEE 1989.
- Stan84 Staniszkis, W., Orlando, S., Rullo, P., Performance-Oriented Database Design Laboratory, w Database Performance State of the Art Report, D.A. Bell (Ed.), Pergamon Infotech Limited 1984.
- Stan90 Staniszkis, W., Database Design w Practical Database Techniques, S. M. Deen (Ed.), Pitman Publishing, 1990.
- PKWB91 ProKit\*WORKBENCH® Reference Manual, McDonnell Douglas Corporation, 1991
- INGR91 Dokumentacja techniczna Ingresu, Ingres Corporation, 1991

# A D / C Y C L E

IBM-mowska filozofia inżynierii projektowania oprogramowania.

Autor: Krzysztof Madrjas, IBM POLSKA

W A R S Z A W A, kwiecień 1992



## Wstęp.

W materiale tym pragnę przedstawić założenia filozofii IBM w zakresie utrzymania spójności oprogramowania użytkowego - dla głównych platform sprzętowych oferowanych przez naszą firmę, poprzez architekturę zwaną Architekturą Systemów Aplikacyjnych (System Application Architecture - SAA). Stale rosnące wymagania dotyczące zwiększenia ilości przetwarzanych informacji wymogły zapotrzebowanie na utworzenie rodziny spójnych systemów operacyjnych, które pozwoliłyby na wykorzystanie jednej aplikacji w różnych środowiskach sprzętowych. Jednocześnie funkcje i dane powinny być rozmieszczone na terenie całego przedsiębiorstwa - w celu zaspokojenia rosnących oczekiwań użytkowników przy minimalnych nakładach na zmiany oprogramowania.

Na podstawie wymagań stawianych w przedsiębiorstwie użytkowym systemom informacyjnym zostaną pokazane możliwości oferowane przez architekturę SAA.

Powyższe sformułowanie definiuje bazę, na tle której działa IBM-owska strategia rozwoju oprogramowania użytkowego dla SAA - AD/Cycle (Application Development - Cykl Rozwoju Aplikacji).

AD/Cycle definiuje funkcje i środowisko poprzez cały cykl rozwoju oprogramowania - od momentu pierwszych rozmów z użytkownikiem do wdrażania i utrzymania działających aplikacji, dzięki zestawowi narzędzi programowych produkowanych przez IBM, użytkowników i firmy współpracujące z IBM w tym zakresie, zintegrowanych na płaszczyźnie projektowania aplikacji.

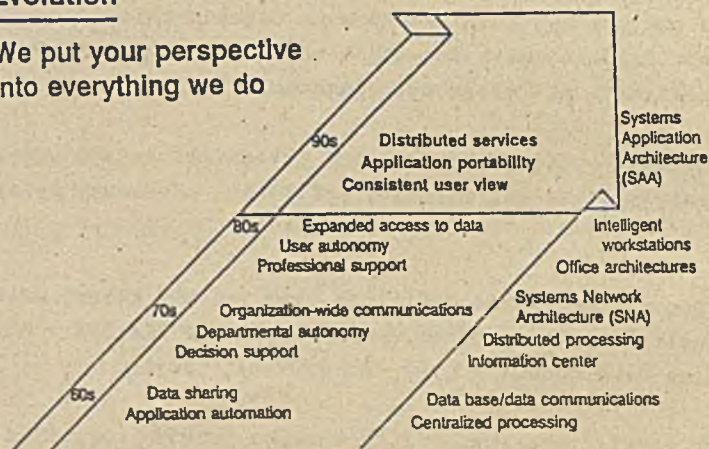
Po przedstawieniu założeń SAA i AD/Cycle przedstawione zostaną wybrane produkty programowe reprezentujące tę strategię tworzenia oprogramowania użytkowego.

## Ewolucja systemów informatycznych.

Przedstawiony poniżej przekrój historyczny przetwarzania informacji od lat sześćdziesiątych do dnia dzisiejszego, pokazuje jak zmieniały się w tym okresie wymagania użytkowników i w jaki sposób nasza firma odpowiadała na nie zmianami architektury i technologii systemów komputerowych.

### Software Strategy: Evolution

We put your perspective  
into everything we do



W latach 60-tych dominowało przetwarzanie zcentralizowane, nastawione na obsługę głównych systemów użytkowych przedsiębiorstwa związanych z rachunkiem ekonomicznym i materiałowym. Wraz ze wzrostem ilości automatyzowanych funkcji rosło zapotrzebowanie na udoskonalenie zasobów pamięci, w której przechowywano informacje i metod dostępu do niej. Odpowiedzią IBM było zaoferowanie baz danych i metod komunikacyjnych - umożliwiających wprowadzenie "transparentności danych". Pozwoliło to na rozwój zasobów pamięci dyskowych użytkowników, przy minimalnych nakładach na modyfikację oprogramowania.



W latach 70-tych użytkownicy zaczęli wykorzystywać informacje w sposób bardziej efektywny, redukując nieaktywny czas pracy aplikacji. Sytuacja ta wymogła konieczność utworzenia oprogramowania wspomagającego podejmowanie decyzji - więc IBM opracował koncepcję Centrum Informacyjnego. Udostępniło ono użytkownikom spójny zestaw narzędzi programowych wspomagających podejmowanie decyzji dla projektantów oprogramowania użytkowego i użytkowników.

Równocześnie u wielu użytkowników powstało zapotrzebowanie na autonomiczne, lokalne przetwarzanie danych, nie wymagające ingerencji centralnego ośrodka przetwarzania w przedsiębiorstwie - było to podstawą wprowadzenia technologii przetwarzania rozproszonego.

Rosnące i coraz bardziej wyrafinowane zapotrzebowania wymagały efektywnego sterowania dostępem do informacji - konsekwencją było opracowanie przez IBM architektury sieci systemów informatycznych (System Network Architecture - SNA) i wprowadzenie "transparentności komunikacji".

Dziś SNA to architektura komunikacyjna, wykorzystywana przez użytkowników do zarządzania komunikacją informacji poprzez różnorodne lokalizacje systemów przetwarzania danych.

W latach 80-tych IBM kontynuował rozwój rozwiązań systemowych poprzednich dekad. Opracowana została koncepcja systemów przetwarzania biurowego - dla umożliwienia tworzenia i dystrybucji dokumentów poprzez całe przedsiębiorstwo. W celu zaspokojenia rosnących zapotrzebowań indywidualnych użytkowników wprowadzony został model inteligentnej (programowalnej) stacji roboczej (Programmable WorkStation) - komputery osobiste. Stały rozwój oprogramowania umożliwił użytkownikom dostęp do profesjonalnych narzędzi programowych.

Obecnie bardziej niż kiedykolwiek nacisk położony jest na obsługę pojedynczego użytkownika na tle całego przedsiębiorstwa. Pracownicy chcą mieć dostęp do informacji i aplikacji - niezależnie od ich lokalizacji. Chcą także wykorzystywać te same funkcje, w taki sam sposób - na różnych systemach komputerowych. Ale równocześnie chcą rozdzielać funkcje na różne systemy w celu optymalnego wykorzystania specyficznych możliwości charakteryzujących różny sprzęt i oprogramowanie systemowe.

Aby wykorzystać krytyczne zasoby w sposób bardziej efektywny, konieczne jest projektowanie aplikacji, które mogą być wykorzystane w różnorodnych środowiskach systemowych i sprzętowych, które mogą przetrwać zmiany w założeniach przy minimalnych zmianach programowych.

Rozwiązania przetwarzania informacji na lata 90-te muszą rozwiązywać totalne potrzeby wszystkich grup użytkowników przedsiębiorstwa - personalne, grupowe, działowe, lokalizacyjne - dostawców i klientów. Celem jest więc opracowanie rozwiązania obejmującego wszystkie te wymagania, pracującego w różnorodnych środowiskach sprzętowych oraz odpowiadającego jednocześnie na różnorodne zapotrzebowania indywidualne użytkowników. Rozwiązanie takie powinno zapewnić transparentny dostęp do danych i aplikacji w przedsiębiorstwie - niezależnie od ich fizycznej lokalizacji.

Dla przykładu, użytkownicy pracujący na jednym poziomie indywidualnie, mający dostęp do danych lokalnych, wykorzystują je i przetwarzają, a następnie rozprowadzają poprzez całe przedsiębiorstwo - wypełniając zawarte w nich polecenia. Takie zastosowanie wymaga zarówno unikatowych zadań jak i wspólnych funkcji, takich jak np. zarządzanie kalendarzem i tworzenie raportów.

Użytkownicy biorą także udział w pracach grupowych, wymagających unikalnych zadań do ich wykonania, lub jednocześnie zadań dzielonych z innymi grupami. Wspólnie grupa tworzy organizację zlokalizowaną w jednym lub wielu miejscach.

Do tego wszystkiego musi być możliwe dzielenie informacji z dostawcami i klientami. Ale nie wszystkie informacje. Dla obsługi wszystkich tych powiązań konieczny jest bardzo sprawny system komunikacyjny. Sieć przetwarzania danych musi zapewnić użytkownikom dostęp do danych i funkcji, których wymagają oraz musi dostarczyć rezultat ich pracy tym, którzy tego oczekują.

Sieć ta musi obsługiwać wiele rodzajów informacji: tekst, grafikę, głos i obraz - integrując różnorodne urządzenia.

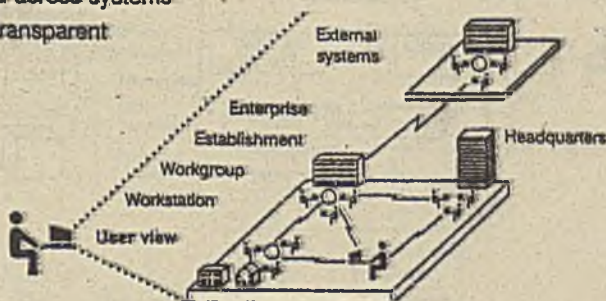


IBM oferuje sprzęt i oprogramowanie dla całego spektrum wymienionych wymagań - proponując koncepcję systemu informacyjnego przedsiębiorstwa (EIS Enterprise Information System).

## Software Strategy: Requirements

### Enterprise Information System

- Distributed across systems
- Location transparent



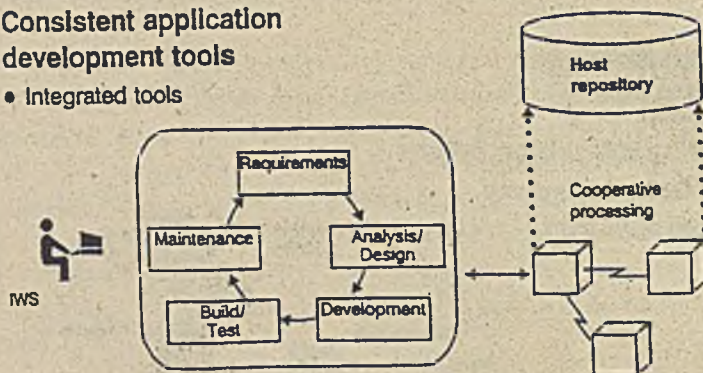
### Środowisko tworzenia aplikacji.

Dla zapewnienia projektantom możliwości tworzenia aplikacji dla systemu informacyjnego przedsiębiorstwa konieczne jest dostarczenie im rodziny zintegrowanych narzędzi programowych do obsługi pełnego cyklu czasu trwania aplikacji. Oznacza to, że narzędzia te muszą pracować w rozproszonym środowisku i tworzyć wspólne zasoby projektowe. Ludzie projektujący, kodujący, testujący i obsługujący oprogramowanie użytkowe wymagają inteligentnych stacji roboczych jako stanowiska pracy. Część funkcji takich jak dzielona składnica (Repository), powinno rezydować w zasobach komputerów centralnych, podczas gdy inne, takie jak np. grafika czy projektowanie są doskonale realizowane na stacjach roboczych. Lokalizacja funkcji powinna być transparentna dla użytkownika.

## Software Strategy: Requirement

### Consistent application development tools

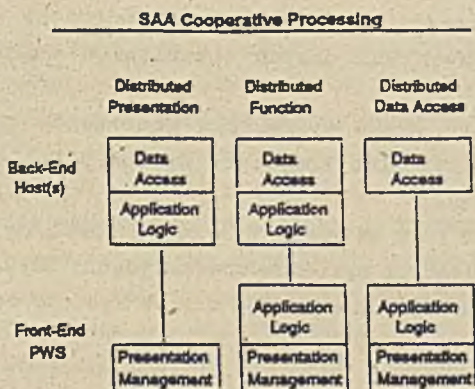
- Integrated tools



Użytkownik systemu informacyjnego musi mieć dostęp do zasobów położonych w dowolnym miejscu objętego siecią informacyjną przedsiębiorstwa lub w sieci zewnętrznej..Tego typu pracę zapewnia przetwarzanie kooperatywne (cooperative processing). Przetwarzanie kooperatywne SAA zapewnia użytkownikowi zintegrowany i prosty dostęp do wszystkich zasobów komputerowych - przez zasoby rozumiemy: dane, sprzęt komputerowy (np. drukarki o dużej wydajności podłączone do systemów centralnych), pakiety programów użytkowych lub specjalne programy niedostępne na pojedynczych stacjach. Użytkownik powinien mieć możliwość wykorzystania tych zasobów bez potrzeby posiadania wiedzy o ich typie i lokalizacji. W tym samym czasie zasoby te muszą być chronione przez specjalne programy zabezpieczające. Przetwarzanie kooperatywne ma jedną postać, chociaż wykonuje się w dwu lub wielu połączonych systemach. Przy tym co najmniej funkcja zarządzania prezentacją graficzną obrazu wykonywana jest na PWS.



Innymi słowy przetwarzanie kooperatywne SAA to szczególna postać przetwarzania rozproszonego, w którym występuje koordynowane wykorzystanie PWS połączonego z jednym lub wieloma komputerami centralnymi dla wykonania transakcji użytkowej. Jak pokazano na rysunku SAA Cooperative Processing, przetwarzanie to może być rozproszone na 3 sposoby:



\* rozproszona prezentacja (distributed presentation) - program użytkowy i obsługa danych wykonywane są w środowisku komputera centralnego. Dane także zapamiętywane są w pamięci tego komputera. Proces aplikacji przegląda dane wykorzystując metody dostępu systemu operacyjnego tegoż komputera. Tylko połączenie z użytkownikiem odbywa się na PWS.

\* rozproszenie funkcji (distributed function) - logika aplikacji wdrożona jest w skoordynowany sposób zarówno na PWS jak i w systemie centralnym. Wykorzystywane dane mogą być usytuowane także w obu środowiskach. Proces aplikacji na życzenie użytkownika odszukuje dane w tej samej lokalizacji dla odzysku i zapisu danych. Funkcja prezentacji wykonywana jest na PWS.

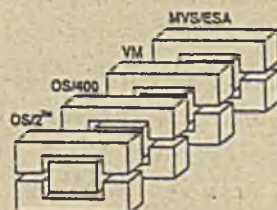
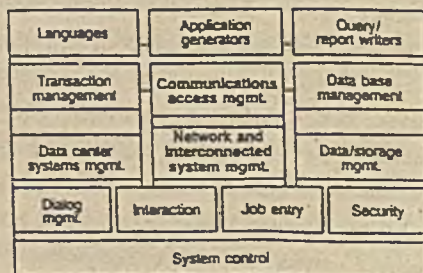
\* dostęp do rozproszonych danych (distributed data access) - cała aplikacja od strony logicznej umiejscowiana jest na PWS. środowisko komputera centralnego może być wykorzystywane jako serwery danych. Jeżeli systemy zarządzania bazą danych (data base management systems DBMSs) w obu środowiskach pracują w sposób kooperatywny, aplikacja na PWS może wykorzystywać dane z zewnętrznych komputerów centralnych poprzez wydanie zapytania do lokalnego DBMS. Lokalizacja i inne szczegóły o danych spoza systemu lokalnego nie muszą być specyfikowane w programie użytkowym na PWS. Ten sposób dostępu do danych został wdrożony w architekturze rozproszonych relacyjnych naz danych (Distributed Relational Database Architecture - DRDA).

Uwaga: w tym przypadku przez środowisko komputerów centralnych jako serwery rozumie się nie tylko komputery System/370 i System/390, ale także AS/400 i inne PS/2.

Ogłoszony w marcu 1987 roku wstęp do SAA zdefiniował interakcyjny model aplikacyjny dla następujących systemów operacyjnych: OS/2, OS/400, VM/CMS MVS/ESA. Komponenty, z których składają się te systemy można przedstawić w trzech grupach.

## Software Strategy: Consistent Software Structure

- Operating system structure extended to multiple systems





Płaszczyzna tworzenia aplikacji i zarządzania komunikacją, zarządzanie bazami danych (identyczne dla wszystkich platform sprzętowych w architekturze SAA) i metodami dostępu do danych oraz płaszczyzna styku ze sprzętem - specyficzna dla każdej platformy sprzętowej, gdyż od niej zależy wykorzystanie charakterystycznych cech poszczególnych typów komputerów.

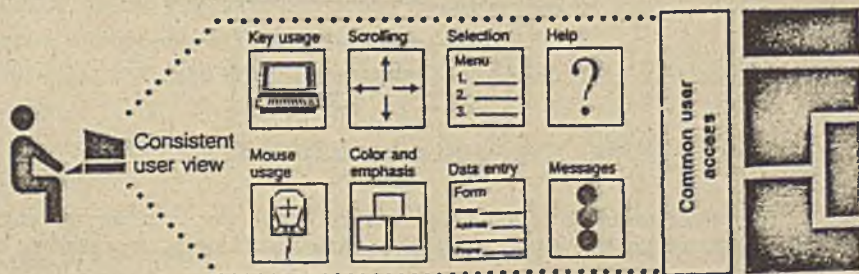
### AD/CYCLE - podstawowe definicje.

Aby umożliwić tworzenie jednorodnych aplikacji architektura SAA wnosi standaryzację podstawowych połączeń w tej rodzinie systemów. Najbardziej znaczące komponenty to: wspólny dostęp użytkownika (Common User Access - CUA), jednolite usługi komunikacyjne (Common Communication Support - CCS) oraz standard połączeń międzyprogramowych (Common Program Interface - CPI). IBM SAA CUA definiuje bazowe elementy komunikacji z użytkownikiem - reguły wspólnej terminologii, wspólnego wyglądu ekranów, identycznych menu i innych interaktywnych technik, które umożliwiają wspólny wygląd i atmosferę dla różnych aplikacji i systemów. Podstawowy cel CUA to ułatwienie obsługi programów i uczenia się.

## Systems Application Architecture: Common User Access

### Developer's perspective

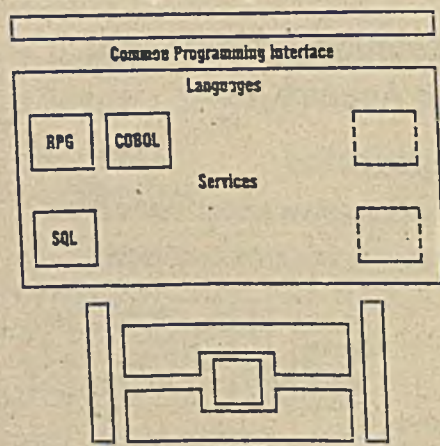
- Defines consistent user view across applications
- Both intelligent and dependent workstations



CPI obejmuje języki programowania, komendy i odwołania wykonywane przez programy. CPI dzieli się na dwie główne kategorie: języki i usługi programowe. Standard ten definiuje założenia, które umożliwiają wykorzystanie aplikacji napisanych w jednym środowisku systemowym w innych systemach operacyjnych.

Podstawowe języki programowania AD/CYCLE to: COBOL, RPG, C, FORTRAN, PL/1, REXX oraz CSP. Zdefiniowane zostały także elementy obsługi baz danych - strukturalny język zapytań (Structure Query Language - SQL) i środowisko zarządzania zapytaniami (Query Management Facility - QMF). Umożliwia to stworzenie jednolitego środowiska obsługi baz danych na wszystkich płaszczynach sprzętowych. CPI definiuje także elementy dialogu pomiędzy użytkownikami a aplikacjami oraz sposób komunikacji pomiędzy programami. Przez wprowadzenie definicji komunikacji dla języków wysokiego rzędu (High Level Language CPI - HLL CPI) umożliwia prostą wymianę danych i funkcji pomiędzy aplikacjami napisanymi w różnych językach.

### Common Programming Elements



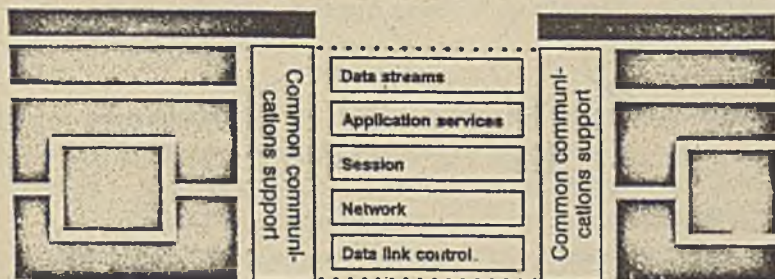
CCS wykorzystywany jest do dołączenia aplikacji, systemów, sieci i urządzeń. Komponent ten definiuje reguły połączeń zgodne z międzynarodowymi



dowymi standardami na różnych płaszczyznach. Podstawowy poziom komunikacyjny to sieci i linie komunikacyjne (protokoły transmisji) wykorzystywane przez wszystkie nasze systemy. Poziom średni to sterowanie sesjami, komunikacja pomiędzy programami, itp. Na najwyższym poziomie znajduje się obsługa strumieni danych oraz usługi aplikacyjne, umożliwiające wykorzystanie usług komunikacyjnych opartych na transmisji zdefiniowanej przez niższe poziomy - w programach aplikacyjnych.

## Systems Application Architecture: Common Communications Support

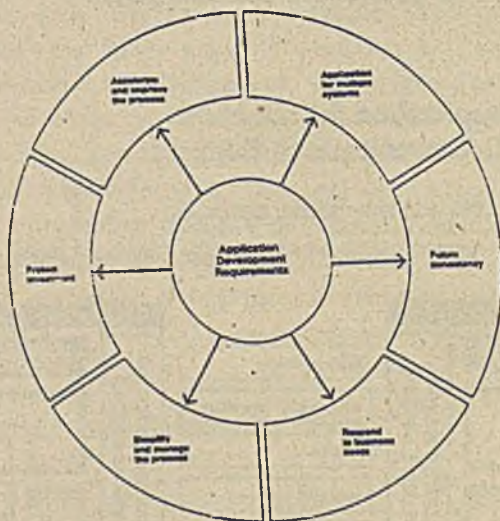
- Established SNA and international standards
- System-to-system/application-to-application communications



Idee AD/CYCLE można zdefiniować przez pytania dobrze znane dzisiejszym projektantom oprogramowania użytkowego:

- co można zrobić w celu przyspieszenia i udoskonalenia środowiska projektowania ?
- jak możemy ochronić nasze wcześniejsze inwestycje (projekty, programy, nakłady pracy) ?
- jak można uprościć i udoskonalic zarządzanie projektowaniem ?
- jak możemy projektować i obsługiwać aplikacje dla różnorodnych systemów ?

- co możemy zrobić w celu adaptacji naszego projektu dla nowych wymagań w przyszłości ?
- jak możemy zapewnić większą elastyczność dla zaspokojenia wymagań użytkowników ?



### Struktura AD/CYCLE.

Struktura AD/Cycle składa się z trzech grup komponentów:

- \* Narzędzia obejmujące cały cykl trwania aplikacji (cross life cycle tools) dostarczają zintegrowanych funkcji, umożliwiających dzięki specjalnym produktom IBM, zarządzanie i sterowanie całym procesem projektowania aplikacji:
  - zarządzanie projektami i procesami
  - analizy impaktowe
  - przygotowanie dokumentacji



- rekonstruowanie procesów i projektów

\* Narzędzia tworzenia aplikacji dostarczają podstawowych funkcji koniecznych przy projektowaniu aplikacji:

- modelowanie danych, procesów, kierunków zmian założeń
- analizy i projektowanie dla baz danych, logiki, ekranów, raportów
- budowa prototypów w celu stworzenie początkowego obrazu aplikacji:
  - + języki i narzędzia do edycji, kompilacji i analizy błędów
  - + generatory, które przenoszą na wyższy poziom projektowanie z poziomu języków trzeciej generacji
  - + systemy baz danych wiedzy, dla projektowania i wspomagania narzędzi projektowych
  - + możliwości testowania i modelowania
  - + wspomaganie obsługi pracy i możliwość rekonstrukcji oprogramowania użytkowego

\* Platforma projektowania aplikacji integruje wszystkie działania związane ze środowiskiem projektowania poprzez:

- CUA - jednolity wygląd i wrażenia estetyczne dla projektantów, niezależność poszczególnych funkcji narzędziowych; stacja robocza AD/Cycle jest zgodna z graficznym modelem SAA CUA
- usługi oferowane przez stację roboczą (workstation services); bazując na standardzie CUA umożliwia użytkownikowi wykorzystanie zaawansowanych możliwości graficznych PS/2
- usługi oferowane przez narzędzia (tools services); zestaw funkcji, których wymagają narzędzia programowe w celu zapewnienia integracji danych i innych wspólnych funkcji
- usługi oferowane przez składnicę (repository services); obsługa składnicy oferuje usługi umożliwiające aktualizację danych, utrzymanie jednolitych poziomów poprawek, itp...
- informacyjny model projektowania aplikacji (application development information model); spójny obraz definicji projektowania systemu użytkowego, takich jak dane, programy, ekrany - łącznie z elementami definicji dostępnych funkcji

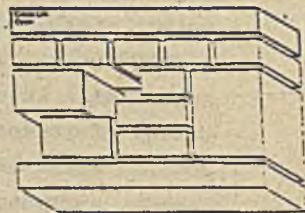
- dwuczęściowy model informacyjny (two part information model)
  - + model ER (entity-relationship = wystąpienie-relacja) dla definicji relacji niskiego poziomu
  - + model obiektowy (object model)

Oprócz wymienionych wyżej platform wyróżniamy dwie grupy produktów:

- \* rodzina produktów podstawowych dla AD/CYCLE - są to produkty SAA obsługi baz danych:
  - DATABASE 2 (DB2)
  - Structured Query Language/ Data System (SQL/DS)
  - Operating System/400 (OS/400) Database
  - Operating System/2 Extended Edition (OS/2 EE) Database
- \* rodzina produktów pokrewnych AD/CYCLE - są to produkty programowe IBM'owskich systemów zarządzania i chociaż nie są one przeznaczone specjalnie dla AD/CYCLE są jednak niezbędne do obsługi produktów podstawowych i bardzo użyteczne w tworzeniu zintegrowanego środowiska:
  - CICS OS/2
  - OS/VS DB/DC Data Dictionary
  - Database Relational Application Directory (DBRAD)
  - Operating System/2 Extended Edition (OS/2 EE)
  - OS/2 Programming Tools and Information Version 1.2
  - Query Management Facility (QMF)

### Przeгляд wybranych produktów AD/CYCLE.

#### Application System (AS) Project Management.



Jest to usługowy program sterujący, który może być wykorzystany do efektywnego zarządzania alokacją zasobów wykorzystywanych przy projekcie. Zawiera on możliwości wykorzystania grafiki i tworzenia raportów dla zilustrowania postępów prac

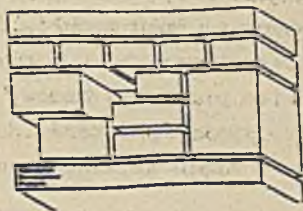
przy opracowywaniu projektu. Można wykorzystywać go zarówno na PWS jak i na komputerach głównych.

ZALETY: Produkt ten pozwala na efektywne planowanie i wykorzystywanie zasobów. Lepsze zarządzanie projektowaniem aplikacji pozwala na zwiększenie wydajności przy projekcie i zmniejszenie kosztów.



## AS/400 Application Development Tools.

Narzędzia Rozwoju Aplikacji AS/400 (ADT) składa się zestawu zintegrowanych narzędzi wykorzystywanych do projektowania, programowania i wdrażania oprogramowania użytkowego na maszynach rodziny AS/400. Programing Development Manager (PDM -



- system zarządzania programowaniem) dostarcza zintegrowane środowisko do wykorzystania narzędzi tworzenia oprogramowania. Umożliwia listowanie obiektów i podejmowanie akcji związanych z wybranymi obiektami.

Source Entity Utility (SEU - program obsługi modułów źródłowych) jest to mocny edytor wykorzystywany przy pisaniu oprogramowania. Pozwala on na kontrolę syntaktyczną i składniową dla następujących kompilatorów: AS/400 PL/1, RPG/400, COBOL/400, AS/400 BASIC i AS/400 Command Language.

Screen Design Aid (SDA - moduł projektowania ekranów) to program wykorzystujący metodę WYSIWYG (What You See Is What You Get - masz to co widzisz) do projektowania ekranów dla środowiska AS/400. Obsługuje on także ekrany projektowane/wykorzystywane w środowiskach systemów S/36 i S/38. Program ten tworzy specyfikacje opisu danych (Data Description Specification - DDS) - zewnętrzne definicje umożliwiające wykorzystanie tak przygotowanych postaci ekranów przez różnorodne aplikacje.

Data File Utility (DFU - moduł obsługi zbiorów danych) jest to proste i łatwe do wykorzystania narzędzie do definicji zbiorów i obsługi prostego wprowadzania danych. Jest to bardzo pożyteczne narzędzie do tworzenia danych testowych.

Advanced Printer Function (AFP - program drukujący o rozszerzonych możliwościach) obsługuje zaawansowane wydruki dla drukarek: IBM 5224 i 5225.

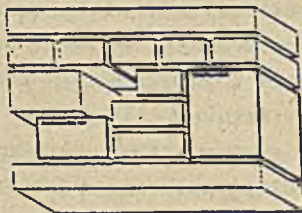
ZALETY: Główna zaleta korzystania z ADT/400 to integracja języków programowania. Dla przykładu PDM wywołuje automatycznie odpowiedni kompilator w zależności od charakterystyki napisanego modułu źródłowego, a SDA generuje ekrany, które mogą być wykonane przez programy wykonane w dowolnym języku. PDM pozwala na definicje własnych komend, które wywołują akcję dla określonych obiektów, a także na modyfikacje systemu pod ściśle określone potrzeby.

SEU zwiększa wydajność pracy programistów dzięki kontroli syntaktycznej zdań modułów źródłowych oraz możliwości generowania zdań charakterystycznych dla danego kompilatora. DFU jest bardzo silnym narzędziem prostocie budowy i obsługi. Umożliwia tworzenie w prosty sposób programów do testowego wprowadzania danych i obsługi zbiorów w ciągu kilkunastu minut.

#### BACHMAN Re-Engineering Product Set.

Zestaw produktów tworzenia i obsługi aplikacji firmy BACHMAN (REPS) składa się z siedmiu komponentów:

- BACHMAN/DATABASE ADMINISTRATOR (DB2)
- BACHMAN/DBA/DB2 Catalog Extract
- BACHMAN/Data Analyst
- BACHMAN/DA Capture (IMS)
- BACHMAN/DA Capture (files)
- BACHMAN/Database Administrator (IDMS)
- BACHMAN/Workstation Manager



Narzędzia te zajmują kilka faz czasu cyklu rozwoju aplikacji. Mogą być one wykorzystywane zarówno przy początkowej fazie rozwoju aplikacji na etapie zbierania zapotrzebowań w przedsiębiorstwie i fazy Analizy/Projektowania, jak i przy projektowaniu oprogramowania i baz danych. Pozwalają one także na nadzór i modyfikacje już eksploatowanych systemów. Charakteryzują się one unikalną cechą - możliwością wykorzystania produkcyjnego systemu użytkowego korzystającego z IMS, IDMS lub DB2 lub zbiorów obsługiwanych przez standardowe metody dostępu jako wejście informacyjne do projektów nowych systemów użytkowych opartych na DB2. Podstawowe funkcje tych narzędzi można przedstawić jako typową sekwencję wykorzystania w następujący sposób:

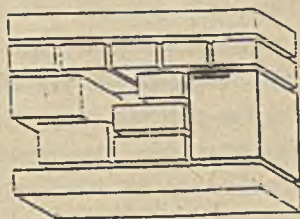
1. BACHMAN/DBA (DB2) Catalog Extract tworzy wyciągi definicji obiektów i statystyki z katalogu DB2 do wykorzystania przez Database Administrator.
2. BACHMAN Database Administrator (DB2) odczytuje definicje DB2 i na ich podstawie tworzy wsad do etapu analizy danych (Data Analyst) oraz postać ER danych do tworzenia definicji DB2 i optymalizacji bazy danych.



3. BACHMAN/DA Capture (IMS) odtwarza definicje IMS i przetwarza je do fazy początkowej, tak aby mógł je wykorzystać Data Analyst.
  4. BACHMAN/DA Capture (files) odtwarza definicje zbiorów dla COBOLu i przetwarza je do postaci akceptowalnej dla Data Analyst.
  5. BACHMAN Database Administrator (IDMS) wykonuje funkcje jak wyżej dla zbiorów IDMS.
  6. BACHMAN/Data Analyst pozwala na graficzną edycję, łączenie i kontrolę danych na poziomie ER dostarczając ekspertowe wspomaganie.
  7. BACHMAN/Workstation Manager integruje funkcje wszystkich produktów rodziny BACHMAN/ERPS przy wykorzystaniu graficznego środowiska okienkowego.
- ZALETY: Korzyści płynące z wykorzystywania narzędzi projektowania bazy danych, modelowania danych, optymalizacji i utrzymania oprogramowania rosną wraz ze stopniem integracji wszystkich tych funkcji. Dzięki tej integracji wyższa jest także jakość oprogramowania użytkowego. Użytkownicy różnorodnych baz danych mogą zmienić swe aplikacje do wspólnego typu. Pozwala to na ochronę inwestycji poniesionych na pracę projektową dzięki możliwości powtórnego przekształcania i modyfikowania (re-engineering) już eksploatowanych aplikacji. Efektem zastosowania tego produktu jest także skrócenie czasu potrzebnego na przygotowanie systemu użytkowego dzięki podniesieniu wydajności pracy oraz zastosowaniu najnowszych technik informatycznych z dziedziny baz wiedzy. Funkcje dostarczane przez te narzędzia można wykorzystywać na PWS z OS/2.

### COBOL Structuring Facility.

Moduł strukturalizacji COBOLu to program umożliwiający modyfikację istniejących programów napisanych przy wykorzystaniu starych kompilatorów do postaci w pełni strukturalnego języka COBOL II. Jeżeli narzędzie to wykorzystywane jest łącznie z modułem konwersji poziomu komend (COBOL/CICS Command Level Conversion Aid - CCCA) program ten wykorzystując techniki sztucznej inteligencji modyfikuje stare oprogramowanie (łącznie z transakcjami konwersacyjnymi) do postaci nowoczesnych strukturalnych modułów. Miejsca niemożliwe do



automatycznej restrukturyzacji są analizowane przy konwersacyjnej współpracy z projektantem/programistą.

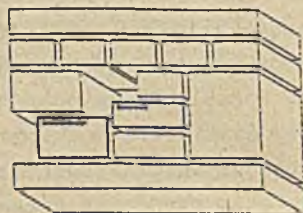
ZALETY: Dzięki zastosowaniu COBOL SF możemy podnieść jakość aplikacji, zwiększyć efektywność projektowania, wykorzystać doświadczenia poprzez:

- zmniejszenie kosztów projektowania i utrzymywania oprogramowania
- eliminacje problemów sterowania rutynami w kodzie niestrukturalnym
- zmniejszenie czasu programisty koniecznego do utrzymania systemu
- przedłużenie życia starych programów
- łatwiejsze tworzenie dokumentacji programowej i projektowej.

### Cross System Product.

CSP to potężny generator aplikacji składający się z trzech komponentów:

- Cross System Product/Application Development (CSP//AD) który umożliwia tworzenie oprogramowania w środowisku System/370, System/390 i PS/2 PWS.
- Cross System Product/Application Execution (CSP/AE) umożliwia wykonywanie transakcji wygenerowanych przez CSP w różnych środowiskach systemowych (nie obejmuje on jednak środowiska IMS/VS)
- Cross System Product/ 370 Runtime Services generuje moduły źródłowe w COBOLu na podstawie modułów CSP oraz umożliwia wykonywanie skompilowanych modułów w środowisku IMS/VS.



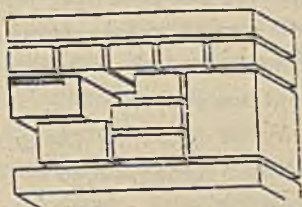
CSP jest to sztandarowy aplikator generacji IBM dla architektury SAA. Produkt ten pozwala na pracę projektantów przy tworzeniu definicji systemu aplikacyjnego na pojedynczych stacjach roboczych lub w środowisku lokalnej sieci teletransmisji (LAN). Dostarcza on wiele możliwości strumieniowego definiowania, testowania i dokumentowania oraz wykonywania wsadowego lub konwersacyjnego nowych systemów użytkowych. Możliwy jest także interakcyjny nadzór i testowe uruchamianie aplikacji. CSP wspomaga tworzenie aplikacji zarówno dla relacyjnych (DB2, SQL/DS, SQL/400) jak i hierarchicznych (IMS/DB, PL/1 DOS/VS) baz danych. Dostarcza także narzędzi do tworzenia prototypowych ekranów i aplikacji.



**ZALETY:** Wykorzystując CSP można zwielokrotnić wydajność profesjonalnych projektantów, jednocześnie rozszerzając możliwości wykorzystania projektowanej aplikacji na wszystkich dostarczanych w ramach SAA płaszczyznach sprzętowych-systemowych. Jako logiczny komponent AD/Cycle CSP umożliwia redukcję kosztów i rozszerzenie możliwości tworzenia oraz modyfikowania systemów użytkowych poprzez uproszczenie procesu rozwoju aplikacji. Zastosowanie możliwości tworzenia prototypów aplikacji na wczesnych etapach projektowania, pozwala na znaczne podwyższenie jakości tworzonych produktów i lepsze usatysfakcjonowanie użytkownika. Dzięki umożliwieniu przetwarzania na głównych systemach IBM - CSP pozwala zabezpieczyć dotychczasowe inwestycje sprzętowe i na rozwój oprogramowania.

#### DevelopMate.

DevelopMate jest projektowane do zastosowań dla analityków i innych osób pragnących wykorzystać do pracy przedsiębiorstwa dane przedstawione z wykorzystaniem modelu ER. Dzięki wykorzystaniu standardowych metod modelowania i zapisywaniu informacji w centralnej składnicy upraszcza komunikację pomiędzy użytkownikiem a profesjonalistami zajmującymi się tworzeniem systemu użytkowego. Komunikacja produktu z projektantem pracującym na PWS została zaprojektowana w sposób umożliwiający uproszczenie i pomoc we wprowadzaniu informacji o modelu działania przedsiębiorstwa w postaci tekstowej. Analityk definiuje dane o działalności, atrybuty i charakterystyki zgodnie z regułami, polityką i dynamiką rozwoju firmy. DevelopMate umożliwia tworzenia z takich danych prototypów aplikacji wykorzystując standardowe ekrany i raporty, z możliwością dokonania dodatkowych modyfikacji przez analityka na stacji roboczej. Taki prototyp pozwala na kontrolę modelu przedsiębiorstwa i sprawdzenie na wczesnym etapie projektu czy spełnia on oczekiwania użytkownika.



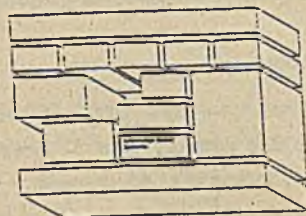
**ZALETY:** Zwykle możliwość sprawdzenia poprawności aplikacji we wczesnym etapie pozwala na znaczne zmniejszenie kosztów i ograniczenie ilości pojawiających się problemów. Dzięki włączeniu sprzężenia z użytkownikiem niemal od początku tworzenia aplikacji, możliwy jest w niemal doskonały

sposób zaspokojenie jego oczekiwań i żądań. Rezultatem jest podwyższenie jakości powstającego systemu i zwiększenie wydajności zespołu tworzącego system użytkowy. DevelopMate dostarcza funkcje będące logicznymi komponentami AD/Cycle i ochraniające nie tylko inwestycje użytkownika we współczesną technologię, ale także dane i programy wyprodukowane przy wykorzystaniu tej technologii.

#### Expert System Environment.

ESE - środowisko systemu ekspertowego może być wykorzystany w różnorodny sposób. Najczęściej program ten jest wykorzystywany do rozwiązywania problemów:

- diagnostycznych
- klasyfikacyjnych
- pomocy na życzenie (help desk)
- doradztwa finansowego



Użytkownicy mogą wykorzystywać ten program bezpośrednio do tworzenia systemów użytkowych - nie wymaga on umiejętności programowania czy projektowania. Wystarczy podać relacje i reguły a system ESE automatycznie utworzy aplikacje. Dzięki standardowym połączeniom ESE może pracować także w środowisku CICS, IMS oraz zapewnia dostęp do relacyjnych baz danych IBM. ESE posiada także możliwości projektowania w środowisku System/370 oprogramowania dla stacji roboczych z IBM DOS lub OS/2.

ZALETY: ESE umożliwia tworzenie aplikacji osobom spoza środowiska informatycznego i dostarcza wiele specjalnych opcji, takich jak:

- proste do wykorzystania funkcje projektowania ekranów
- możliwości graficznego obrazowania danych i tworzenia histogramów.
- proste do wykorzystania metody dostępu do danych zapamiętanych w relacyjnych bazach danych (DB2 i SQL/DS)

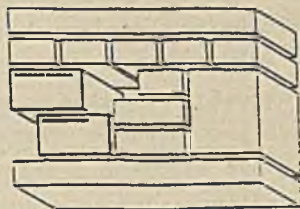
ESE pomaga wykorzystać wysoko produktywne narzędzia tworzenia aplikacji łącznie z możliwością tworzenia prototypów. Dzięki uproszczeniu nadzoru nad pracującymi aplikacjami ESE umożliwia zmniejszenie kosztów utrzymania oprogramowania użytkowego.



## Index Technology Tools.

Narzędzia firmy Index Technology do automatyzacji dwóch pierwszych faz cyklu modyfikacji oprogramowania użytkowego składa się z następujących modułów:

- PC Prism
- Excelerator/IS
- Customizer
- Excelerator dla DB2
- Excelerator dla CSP/AD



PC Prism pomaga w tworzeniu modelu przedsiębiorstwa i przepływie danych. Excelerator/IS tworzy całkowity warsztat do analiz i projektowania.

Index Technology rozszerzyła możliwości Exceleratora dostarczając wspomaganie dla specyficznych produktów IBM. Excelerator dla DB2 umożliwia modelowanie i projektowanie baz danych DB2. Excelerator dla CSP dostarcza środowisko do projektowania aplikacji specyficznych dla CSP z dwudrożnym łącznikiem poprzez zewnętrzny format źródłowy. Przy wykorzystaniu Excelerator'a dla CSP i CSP można utworzyć kompletną aplikację, która następnie może być wykonywana w środowisku CSP.

Customizer pozwala na dostosowania parametrów środowiska projektowania do specyficznych wymogów danego przedsiębiorstwa. Moduł ten może być wykorzystany do modyfikacji słowników, i do tworzenia nowych relacji, unikalnych menu, grafiki, ikon, itp.

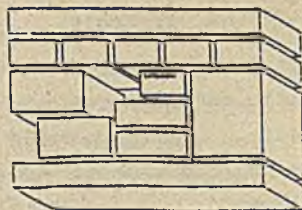
Excelerator dostarcza także narzędzi pomocniczych do tworzenia prototypów. Także ekrany mogą być projektowane w czasie projektowania.

**ZALETY:** Narzędzia CASE Index Technology oferują różnorodne korzyści. Jeżeli projektant może w prosty sposób modyfikować środowisko swej pracy rezultatem jest pełne wykorzystanie czasu pracy i możliwości dla osiągnięcia optymalnych efektów. Użytkownik może nie tylko wpływać na modyfikację środowiska tworzenia aplikacji, ale także wykorzystywać wielokrotnie technologie, wybierając optymalne rozwiązanie dla najlepszej jakości produktu końcowego. Dzięki wykorzystaniu Excelerator dla IBM DB2 i Excelerator dla CSP/AD tworzący aplikację ma możliwość integracji kilku faz cyklu modyfikacji aplikacji.

## Inspect.

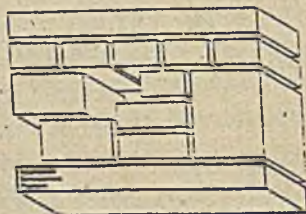
INSPECT wspomaga interaktywne lub wsadowe testowanie i analizy błędów dla programów napisanych w językach: C/370 i PL/1, włączając aplikacje napisane w obu językach jednocześnie. Testowane programy mogą być śledzone z dokładnością do poszczególnych zdań źródłowych modułu. Możliwe jest modyfikowanie zmiennych oraz definiowanie nowych, zmiany w logice programu. Przy wykorzystaniu wielokrotnych okien wszystkie informacje podawane są w bardzo prosty i klarowny sposób. Możliwe jest asynchroniczne modyfikowanie prędkości pozwalające na indywidualne strojenie środowiska testowego.

ZALETY: Testowanie i usuwanie błędów z nowoczesnych aplikacji często jest żmudnym zajęciem, pochłaniającym dużo czasu, zwłaszcza jeżeli programy użytkowe napisane są w więcej niż jednym języku. INSPECT to nowoczesny, bardzo mocny produkt IBM. Oszczędza on czas programistów konieczny do uruchomienia i testowania nowych programów - dzięki rozwiązywaniu problemów wynikającym z błędów w programach użytkowych szybko i efektywnie. Produkt ten dostarcza technologii umożliwiającej zwiększenie wydajności projektowania i programowania - jest to prosty w wykorzystaniu logiczny komponent AD/CYCLE.



## Repository Manager/MVS.

RM - repository (składnica) może stać się pojedynczym miejscem składowania informacji dotyczących projektowania aplikacji w przedsiębiorstwie. Informacja jest zapamiętana tylko jeden raz i tylko w jednym miejscu, a dostęp jest dzielony przez wszystkie narzędzia stosowane do projektowania i modyfikacji oprogramowania użytkowego poprzez cały cykl jego trwania. Wykorzystanie modelu ER (entity-relationship = wystąpienie-relacja) i technik modelowania do opisu danych o informacji w składnicy, tak jak i specyfikacji charakterystyk logiki aplikacji dostarcza zaawansowanych możliwości modelowania.





Repository Manager dostarcza także obsługę łączników obiektowych, gdzie dane i związane z nimi akcje zapamiętane są jako typy obiektów i związane z nimi metody. Narzędzia składnicy mogą pobierać informacje ze składnicy odwołując się do danych z pomocą usług ER lub usług obiektowych. Dane obiektowe mogą być zapamiętane wewnątrz składnicy lub poza nią, a informacje z informacjami kontrolnymi zapamiętanymi w składnicy.

RM pozwala na kontrolowanie zmian zapamiętanego modelu informacyjnego i sterowanie dostępem do informacji zdefiniowanych przez ten model. Przez usługi obiektowe możliwy jest także kontrolowany dostęp do informacji rezydujących w bibliotekach i innych zbiorach. Repository Manager/MVS jest implementacją SAA Repository CPI w środowisku ESA. CPI obsługuje zbiór definicji, manipulację i sterowanie informacjami zapamiętanymi w składnicy.

ZALETY: RM umożliwia utworzenie centralnego punktu sterowania i kontroli dla wszystkich informacji o przedsiębiorstwie i projektowanej aplikacji. Większość tych danych przechowywana jest w składnicy. Dane zapamiętane w odseparowanych (zewnętrznych) zbiorach, takie jak np. moduły ładowalne mogą także pozostawać pod kontrolą RM. Produkt ten wspomaga tworzenie modelu danych przedsiębiorstwa, który zawiera wszystkie niezbędne do budowy aplikacji dane. Model ten pozwala na zwiększenie wydajności tworzenia aplikacji poprzez:

- redukcję redundancji danych
- redukcję problemów definicji danych
- tworzenie modeli prototypowych

RM jest jednym z głównych komponentów logicznych AD/Cycle - pozwala na dzielenie informacji zawartych w składnicy przez narzędzia.

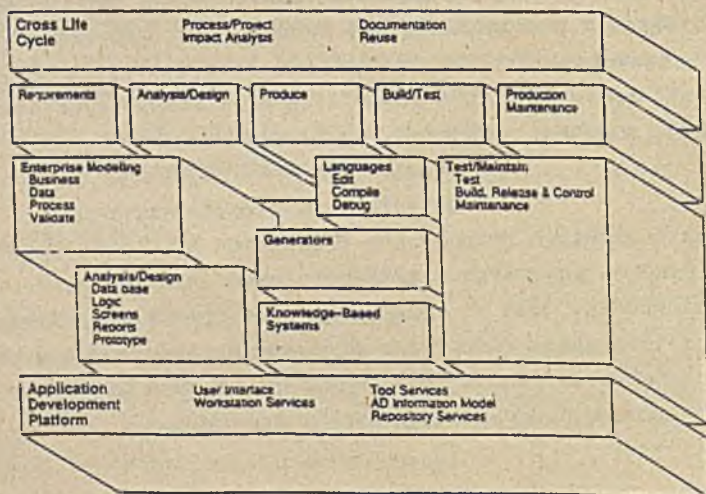
Informacja wprowadzana jest do składnicy tylko jeden raz, a następnie wykorzystywana jest przez różnorodne programy narzędziowe. RM zapewnia także bezpieczeństwo danych, a integracja danych jest prosta do osiągnięcia dzięki jednemu centralnemu punktowi kontroli.

#### Podsumowanie.

Ogłoszenia SAA i AD/Cycle przyświecała nie chęć dostarczenia gotowych do zastosowania narzędzi tworzenia i modyfikacji oprogramowania użytkowego, lecz stworzenia szkieletu systemu tworzenia aplikacji przez zdefiniowanie standardów połączeń i podstawowych definicji.

Szkielet ten, to fundament budowy narzędzi do tworzenia aplikacji. Poprzez ogłoszenie założeń IBM zaproponował współpracę wszystkim firmom tworzącym oprogramowanie. Charakterystycznym zjawiskiem dla tej dziedziny jest fakt, że wiele spośród omówionych tu produktów pochodzi nie z IBM, a z firm będących partnerami naszej firmy przy tworzeniu oprogramowania AD/Cycle. Firmy te są objęte wspólną nazwą International Alliances.

SAA obejmuje tylko niektóre płaszczyzny sprzętowe oferowane przez naszą firmę. Dla systemów nie objętych SAA, tzn VSE i AIX nie jest oferowane jest obecnie tak kompleksowe rozwiązanie. Jednak już dziś wiele z produktów oferowanych dla systemów objętych SAA może także pracować pod kontrolą VSE lub AIX. Także wiele standardowych złączy programowych ma już dziś swoje odpowiedniki w tych systemach - co umożliwia prosty transfer danych i zbiorów. Dalsza integracja tych środowisk będzie postępowała wraz z upływem czasu.



Sądzę, że można powiedzieć, że AD/Cycle wytycza drogę w przyszłość. Przyszłość, w której użytkownik nigdy nie będzie sam - zawsze będzie mu towarzyszył IBM.



Zestawienie wykorzystanych w tekście nazw zastrzeżonych.

IBM, Personal System/2, PS/2, Application System/400, AS/400, C/2, COBOL/2, Database 2, DB2, MVS, MVS/ESA, MVS/XA, Operating System/400, OS/400, OS/2, Presentation Manager, QMF, Repository Manager, RPG/400, SQL/400, System/37, System/370, System/38, System Application Architecture, SAA, VM, VM/ESA, VM/XA są nazwami zastrzeżonymi International Business Machines Corporation.

BACHMAN Re-Engineering Product Set, BACHMAN/Database Administrator(DB2), BACHMAN/Database Administrator (DB2), BACHMAN/DataAnalyst, BACHMAN/Data Capture(IMS), BACHMAN/Data Capture(Files), BACHMAN/DBA (DB2), Catalog Extract, BACHMAN/Workstation Manager są nazwami zastrzeżonymi Bachman Information Systems, Inc. .

Customizer, Excelerator PC/Prism, Excelerator for DB2, Excelerator for CSP są nazwami zastrzeżonymi Index Technology Corporation.

IDMS jest nazwą zastrzeżoną ICulinet Software, Inc.

Zestawienie wykorzystanych w opracowaniu skrótów i nazw obcojęzycznych.

System Application Architecture (SAA - architektura systemów aplikacyjnych).

Application Development (AD/Cycle - cykl rozwoju aplikacji).

System Network Architecture (SNA - architektura sieci systemów informatycznych )

Programmable WorkStation (PWS - inteligentna (programowalna) stacja robocza )

Enterprise Information System (EIS - systemu informacyjnego przedsiębiorstwa )

Repository (składnica)

distributed presentation (rozproszona prezentacja)

distributed function (rozproszenie funkcji)

distributed data access (dostęp do rozproszonych danych )

data base management systemes. (DBMSs - systemy zarządzania bazą danych)

Ditributed Relational Database Architecture ( DRDA - architekturze rozproszonych relacyjnych baz danych)

Common User Access ( CUA - wspólny dostęp użytkownika )

Common Communication Support ( CCS - jednolite usługi komunikacji )

Common Program Interface (CPI - standard połączeń międzyprogramowych )

Structure Query Language (SQL - strukturalny język zapytań )

Query Management Facility (QMF - środowisko zarządzania zapytaniami )

High Level Language CPI (HLL CPI -komunikacji dla języków wysokiego rzędu )

Cross Life Cycle Tools ( Narzędzia obejmujące cały cykl trwania aplikacji )

repository services (usługi oferowane przez składnicę )

workstation services (usługi oferowane przez stację roboczą )

toole services (usługi oferowane przez narzędzia )

application development information model (informacyjny model projektowania aplikacji )

two part information model (dwuczęściowy model informacyjny )

model entity-relationship (model ER - model wystąpienie-relacja)

object model ( model obiektowy )



AS/400 Application Development Tools (ADT - Narzędzia Rozwoju Aplikacji AS/400 )

Source Entity Utility (SEU - program obsługi modułów źródłowych)

Screen Design Aid (SDA - moduł projektowania ekranów)

What You See Is What You Get (WYSIWYG - masz to co widzisz)

Data Description Specification (DDS - specyfikacje opisu danych )

Data File Utility (DFU - moduł obsługi zbiorów danych)

Advanced Printer Function (AFP - program drukujący o rozszerzonych możliwościach)

Data Analyst ( moduł analizy danych )

COBOL/CICS Command Level Conversion Aid ( CCCA - modulem konwersji poziomu komend )

Cross System Product/Application Development CSP//AD)

Cross System Product/Application Execution (CSP/AE)

Local Area Network (LAN - lokalna sieć teletransmisji)

Expert System Environment (ESE - środowisko systemu ekspertowego)

# LISTA SPONSORÓW

Sponsorami Piątej Wiosennej Szkoły PTI są:

Ascom Zelcom AG

Représentation  
Suisse Romande  
8, Rue Saint-Joseph  
CH-1227 Genève-Carouge  
Téléphone 022 - 43 16 82

**ascom** *votre partenaire dans le domaine des transmissions des données et de la parole.*

Warszawa tel/fax: +48 2 6437051

**digital**

Equipment – Polska

Al. Niepodległości 190  
00-618 Warszawa, Polska  
Tel (22) 25 77 85  
Telex (22) 25 77 86

Case Pol

ZBIGNIEW ANTONI PIOTROWSKI

BSc (Comp) UNSW, Australia

02-605 Warszawa

Ursynowska 60 m 7

tel. 44 73 79

fax. 44 75 54

oraz

**IBM Polska**

ul. Złota 44/46, Pewex Building, PL 00-120 Warszawa

Adres do korespondencji / Mail Address

skr. 399, 00-950 Warszawa 1

**CSBI**

**COMPUTER  
SYSTEMS FOR  
BUSINESS  
INTERNATIONAL LTD.**

Spółka z o.o.

02-119 Warszawa

ul. Pruszkowska 17

KOMERTEL 39-12-04-48

TELEFON 659-04-85

659-04-16

TELEFAX 659-04-85

TELEX # 825085 CSBI

**CENTRUM INFORMATYKI  
ZETO-RODAN Sp. z o.o.  
CZERNIAKOWSKA 73/79  
00-718 WARSZAWA  
POLAND**

**RODAN**

Phone INTERNATIONAL ++48 39 120370

Fax INTERNATIONAL ++48 39 120370

Phone: 22-412181

Fax: 22-412184

PBK IX o / Warszawa 370031-35103-136



## Digital Equipment Corporation firma otwartych możliwości

\* Digital Equipment Corporation, czyli DEC jest wiodącym światowym dostawcą sieciowych systemów komputerowych, oprogramowania oraz usług. Digital produkuje w integracji systemów informatycznych pochodzących od różnych producentów. Jest firmą o zasięgu międzynarodowym i ponad połowę swych operacji dokonuje poza granicami Stanów Zjednoczonych, wytwarzając i promując nowe produkty oraz zapewniając profesjonalną obsługę klienta w obu Amerykach, Europie i w rejonie Pacyfiku.

Digital oferuje sprzęt i oprogramowanie w pełnym zakresie poczynając od komputerów przenośnych, poprzez pecety biurowe, stacje robocze, serwery, systemy typu "mainframe" oraz oprzyrządowanie sieciowe umożliwiające budowanie lokalnych i rozległych sieci komputerowych. Tysiące aplikacji działających na tak szerokiej bazie sprzętowej i systemowej obejmują przetwarzanie transakcyjne, bazy danych, telekomunikację, bankowość i finanse, sterowanie i zbieranie danych w czasie rzeczywistym, obliczenia macierzowe, edukację, prace edytorskie, zarządzanie produkcją, ochronę zdrowia i narzędzia programowania.

Założony w 1957 roku, Digital postawił na intensywne badania i rozwój w zakresie sieci komputerowych, stając się właścicielem największej cywilnej sieci komputerowej na świecie. Dzięki innowacyjności, Digital zajmuje czołową pozycję na rynku komputerów o otwartej architekturze spełniającej określone standardy, które pozwalają łączyć systemy cyfrowe VAX/VMS oraz RISC/UNIX. Firma bierze udział w wielu międzynarodowych inicjatywach i organizacjach, których celem jest wypracowanie standardów umożliwiających integrowanie systemów składających się z elementów oferowanych przez różnych producentów.

### Digital Equipment Corporation w Polsce

Digital Equipment Corporation był już znany w Polsce w latach siedemdziesiątych dzięki rodzinom komputerów PDP-11 i VAX-11/780 oraz systemom operacyjnym RSX-11 i VMS. Ze względu na zaawansowaną technologię wszystkie wyroby Digitala zostały w przełomowych latach 80-dziesiątych objęte embargiem. 20 stycznia 1992 roku Digital oficjalnie powrócił i rozpoczął działalność w Polsce.

Przez ten czas zmieniła się Polska, zmienił się także Digital. W ciągu tych lat zebrano wiele nowych doświadczeń, zwiększyły się umiejętności specjalistów, powstały nowe systemy. Najważniejsza zaś zmiana polega na innym podejściu firmy do współpracy z klientem. W Digitalu istnieje bowiem świadomość w jak wielkim stopniu informatyka na świecie i w Polsce opanowała nowe obszary zastosowań i jak wielu nowych użytkowników zaczęło rozumieć i praktycznie wykorzystywać zalety komputerów. Z drugiej jednak strony stały rozwój technologii informatycznych spowodował taką wewnętrzną złożoność struktur systemów, że do ich zaprojektowania konieczna jest współpraca wielu specjalistów. Dlatego też najważniejszym zadaniem dla Digitala stało się takie zorganizowanie działalności, aby możliwe było poznanie potrzeb i oczekiwań klientów, a następnie zaproponowanie procedur, które dokładnie by je realizowały.

Nowa filozofia obsługi klienta została z sukcesem sprawdzona w USA i w Europie. Jej sprawna realizacja wymaga istnienia wysokiej klasy ekspertów znających zarówno środowisko i problemy klienta, jak też możliwości produktów Digitala. Dlatego w Polsce zatrudniono zespół polskich informatyków i specjalistów z

różnych dziedzin, a następnie przeszkolono ich w ośrodkach Digitala. Jednocześnie mogą oni korzystać z firmowej sieci komputerowej zbudowanej z 80 tysięcy węzłów na całym świecie. Bezpośredni dostęp do infrastruktury firmy pozwala w każdym momencie uzyskać rzetelną informację oraz zapewnić klientowi wysoki poziom obsługi.

Digital w swoim planie wieloletniej działalności na terenie Polski zamierza dokonywać znacznych inwestycji umożliwiających zbudowanie sprawnie funkcjonującej organizacji oferującej bogaty pakiet usług, oprogramowania i sprzętu komputerowego oraz współpracującej z wieloma polskimi organizacjami państwowymi i prywatnymi. Partnerem Digitala na terenie Polski może stać się firma wykorzystująca jego systemy, opracowująca specjalistyczne aplikacje, nowe oprogramowanie oraz przygotowująca analizy potrzeb klientów. Również dla partnerów Digital oferuje odpowiednie przeszkolenie i pomoc merytoryczną.

Digital zdaje sobie sprawę z podstawowych potrzeb Polski w zakresie unowocześniania technik informatycznych. Firma stawia do dyspozycji swoich specjalistów, którzy wspólnie z zespołami klientów wypracowują najlepsze sposoby modernizacji infrastruktury informatycznej w Polsce.

## Otwieranie nowych możliwości

Coraz poważniejszym problemem staje się integracja systemów komputerowych pracujących pod różnymi systemami operacyjnymi. Systemy te pojawiały się w przedsiębiorstwach w miarę powstawania nowych problemów, które miały być rozwiązywane za ich pomocą. Na przykład firma, która uznała, że komputery klasy PC są dla niej wystarczające, po pewnym czasie stwierdza, że niezbędny staje się komputer silniejszy, o większej pamięci lub szybkości. Zwykle te komputery nie mają możliwości współdziałania, a co gorsze trudno jest przenieść działające aplikacje z jednej platformy sprzętowej na drugą. Z tego powodu duży wysiłek wkładany w rozwój systemu informatycznego przedsiębiorstwa może być zmarnowany. Problem integracji dotyczy nie tylko różnych platform sprzętowych i systemowych, ale także odmiennych sposobów przetwarzania, przechowywania i prezentowania danych. Systemy umożliwiające działanie aplikacji w środowisku sieciowym na różnych platformach sprzętowych i systemowych nazywamy systemami otwartych możliwości (Open Systems).

Otwartość stanowi istotę strategii i stylu prowadzenia działalności Digitala nie tylko w zakresie rozwiązań sprzętowych i programowych, ale także układów partnerskich i proponowanego serwisu. Forsowana obecnie przez Digital idea systemów otwartych możliwości (Open Systems) daje klientom swobodę wyboru najlepszych aplikacji po najkorzystniejszej cenie działających na prawie każdej popularnej platformie sprzętowej i systemowej. Digital już obecnie proponuje rozwiązanie problemu integracji systemów komputerowych poprzez standaryzację wszelkich interfejsów z otoczeniem tej aplikacji. Pakiet programowy realizujący usługi integracyjne nazywa się NAS (Network Application Support) lub bardziej swojsko WAS (Wspomaganie Aplikacji Sieciowej).

NAS definiuje pewien model interakcji aplikacji zwany dialogiem aplikacji. Dla każdego dialogu określono zbiór standardów, które muszą obowiązywać w ramach tego dialogu. W modelu realizowanym w ramach NAS zdefiniowano cztery podstawowe dialogi, które może prowadzić aplikacja. Są to:

- dialog z użytkownikiem;
- dostęp do danych i informacji;
- komunikacja z innymi aplikacjami;
- współpraca z systemem operacyjnym.



W tradycyjnym oprogramowaniu każdy dialog aplikacji jest uzależniony od systemu, na którym działa. W architekturze NAS jest on odizolowany od konkretnej platformy. Pakiety realizujące usługi NAS mogą być wkomponowane w inne produkty programowe Digitala lub występują w postaci osobnych modułów o nazwach NAS 200, NAS 250, NAS 300 lub NAS 400.

W obecnej chwili pakiety NAS pozwalają integrować systemy pracujące na różnych platformach sprzętowych i systemowych pochodzących od wielu producentów, a w szczególności: VMS/VAX, ULTRIX/RISC, DOS, OS/2, SunOS czy Apple Macintosh. W przyszłości krąg komputerów i systemów integrowanych w ramach NAS znacznie się rozszerzy. Równocześnie zostanie rozszerzony zbiór usług oferowanych przez NAS. Do tej pory powstało setki produktów, które w pełni wykorzystują środowisko NAS, a ich liczba stale rośnie. Digital zapewnia kontynuację strategii związanej z integracją systemów gwarantując, że nakłady ponoszone na rozwijanie aplikacji dzisiaj nie zmarnują się w przyszłości.

## Digital na czele konkurencji

Druga firma komputerowa świata Digital Equipment Corporation (DEC) w drugiej połowie 1991 roku ruszyła do ataku na wielu frontach. Po oficjalnym otwarciu oddziałów na Węgrzech i Czechosłowacji Digital utworzył oddziały w Polsce i Rosji. Planuje także otwarcie oddziałów na Ukrainie, Litwie, Łotwie i w Estonii.

W kwietniu ubiegłego roku cały świat obiegła wiadomość o zainicjowaniu przez Digital, Mips, Microsoft, SCO i Compaq inicjatywy ACE (Advanced Computing Environment), której celem jest połączenie platform sprzętowych i systemowych bazujących na procesorach Intel i MIPS. Inicjatywa przyjęta sceptycznie w zeszłym roku obecnie grupuje już ponad 100 firm chcących pracować nad rozwijaniem otwartego środowiska sieciowego. W roku 91 Digital związał się z wieloma partnerami, z których najważniejsi to Microsoft, Apple, Cray, Philips czy MassPar Computer Corp. Oznacza to szerokie otwarcie firm dla układów partnerskich, które mają dzisiaj decydujące znaczenie dla powodzenia inicjatyw o globalnym zasięgu.

Dla uważnego obserwatora wszystkie działania Digitala wiązały się w logiczną całość. Widać było, że firma wyraźnie zmierza do określenia podstawowych platform sprzętowych i systemowych, które służyłyby jej do budowy otwartych sieci komputerowych. Dlatego wiadomość dotycząca konstrukcji nowego procesora Alpha, którą podał Digital 25 lutego do publicznej wiadomości nie powinna być zaskakująca. Różne informacje docierające z Digitala w ciągu ostatnich lat tylko potwierdzały starania firmy zmierzające do konstrukcji nowego procesora, który w przyszłości zastąpiłby zasłużone procesory VAX. Alpha z całą pewnością zapewni taką ciągłość.

## Co dalej?

Wydaje się, że Digital posiadając obecnie tak wyrafinowaną technologię oraz systemy komputerowe na procesorach VAX i MIPS, które w każdej kategorii mogą z powodzeniem konkurować z wyrobami innych producentów nie musi natychmiast wprowadzać na rynek systemów bazujących na procesorze Alpha. Co więcej, można się spodziewać, że niektórzy z konkurentów mogą mieć problemy z przeskoczeniem w odpowiednim czasie poprzeczki, którą wysoko postawił Digital. Dlatego procesor Alpha należy traktować przede wszystkim jako inwestycję przyszłościową, która powinna się rozwijać własnym tempem nie konkurując w okresie przejściowym z systemami opartymi o procesory VAX i MIPS. Niemniej Digital założył 30 centrów, w których zainteresowani użytkownicy i firmy programistyczne będą mogli już dzisiaj przenosić programy działające w środowisku systemów VMS, ULTRIX i OSF/1 na procesor Alpha. Przenoszenie

programów polega na tłumaczeniu programów binarnych za pomocą specjalnego translatora. Przetłumaczony w ten sposób program jest akceptowany przez procesor Alpha. W tej sytuacji trzeba raczej założyć, że Alpha, na którą będzie powszechnie udzielana licencja powinna stanowić jeszcze jedną przyszłościową platformę sprzętową dla inicjatywy ACE.

Digital z 34-letnią historią w produkcji sprzętu i oprogramowania, ze 120 tysiącami pracowników rozmieszczonymi na całym świecie i z obrotami 13,9 mld. dolarów ma wystarczający potencjał aby z powodzeniem inwestować na polskim rynku w celu uzyskania w niedalekiej przyszłości rzetelnie wypracowanych zysków.

### Alpha - szczegóły techniczne

Pierwsze wiadomości o próbach konstrukcji własnego procesora typu RISC pochodzą z Digitala z 1982 roku. Po siedmiu latach firma dysponowała procesorem 32-bitowym napędzanym 50 MHz zegarem. Jednakże te procesory nie były tak zaawansowane technologicznie aby zapewniać odpowiednią przewagę w stosunku do konstrukcji konkurencyjnych oraz kilkunastoletni horyzont czasowy potrzebny dla ich rozwoju. Alpha wydaje się spełniać te założenia.

Alpha jest procesorem 64-bitowym typu RISC wykonanym w 0,75 mikronowej technologii CMOS. W układzie zasilanym napięciem 3,3 V i zużywającym przeciętnie 23 W, którego wymiary wynoszą 16,8 na 13,9 mm mieści się około 1,7 mln. tranzystorów. W warunkach laboratoryjnych procesor jest napędzany zegarem 200 MHz, co według ostrożnych oszacowań powinno zapewniać przewagę 1,5 roku nad innymi konkurentami. Robocza wersja działająca z zegarem 150 MHz, wykonująca maksymalnie 300 milionów rozkazów na sekundę (MIPS) będzie w sprzedaży w połowie roku. Architektura procesora oznaczonego symbolem 21064 zapewnia dwupotokową realizację rozkazów. Oprócz stałoprzecinkowej jednostki arytmetycznej w tym samym układzie znajdują się: potokowa jednostka zmiennego przecinka, 8 32-bitowych rejestrów dla danych i rozkazów, umożliwiające przesłania z szybkością 3,2 GB/s oraz jednostka zarządzania pamięcią.

Digital posiada 10-letni plan rozwoju Alpha, w którym przewidzianych jest kilka wersji procesora. Rozwój technologii powinien zapewnić zwiększenie upakowania układu do 4 mln. tranzystorów oraz współpracę z zegarem 300 MHz. Obok głównego zespołu konstrukcyjnego, zostały wylonione dwie grupy, z których pierwsza będzie realizować procesor wykonywany w arsenu galu o szybkości działania do 1000 MIPS, zaś druga zajmie się optymalizacją kosztów wytwarzania procesorów używanych do budowy tańszych konfiguracji. Grupa inżynierów technologów liczy 500 osób, które działają w Hudson (Massachusetts) w USA oraz w South Queensferry w Szkocji.



**ascom**

**m y ś l i   d a l e j**

Głównymi obszarami działalności koncernu Ascom są dynamicznie rozwijające się na całym świecie dziedziny: telekomunikacja i automatyzacja usług. W ramach tych technologicznie wysoko rozwiniętych dziedzin zostały otwarte granice, ujednolicona się standardy, a Ascom ze swoją główną siedzibą w Szwajcarii i filiami w krajach Europy, Ameryki Północnej i Azji zajmuje silną międzynarodową pozycję.

# ascom

# na właściwej drodze

## Dział: **sieci publiczne** ( Public Networks )

oferuje budowę nowych i rozbudowę istniejących już systemów telekomunikacyjnych obejmujących sieci telefoniczne jak również sieci umożliwiające tworzenie baz danych oraz łączność mobilną. W tej sferze swojej działalności koncern zdobył mocną pozycję na szwajcarskim rynku i wciąż wzrasta jego aktywność oraz znaczenie poza Szwajcarią w dziedzinach: transmisji i technicznych środków wymiany informacji. Działalność w zakresie transmisji jest prowadzona w formie joint venture z firmą Ericsson, pod kierownictwem Ascom.

## Dział: **sieci prywatne** ( Enterprise Networks )

jest jednym z najbardziej znaczących światowych oferentów prywatnych sieci telekomunikacyjnych o zasięgu międzynarodowym. Jego rdzeń tworzy wiodąca w tej dziedzinie amerykańska firma Timeplex Inc., którą koncern pozyskał w 1991 roku. Dział Enterprise Networks oferuje ponadto sieci specjalne przeznaczone do automatycznego alarmowania i mobilizacji oraz wspomaganie sieci i systemów energetycznych. W zakresie sieci prywatnych Ascom prowadzi także różnorodną działalność w krajach Europy Wschodniej. Krok milowy tej działalności stanowi tworzona obecnie dla Czecho - Słowackich Kolei sieć X.25. Dział Enterprise Networks wraz z działem usług i dystrybucji oferuje różnorodne usługi w zakresie komunikacji, a w szczególności:

- systemy kablowe
- sieci lokalne ( LANs )
- sieci metropolitalne ( MANs )
- sieci rozległe ( WANs )
- systemy zarządzania siecią



**Dział: Środki łączności osobistej**

( Personal & Business Communications )

oferuje produkty, systemy i usługi umożliwiające komunikację werbalną i przekazywanie danych takich jak: kablowe i bezprzewodowe telefony oraz systemy jak również urządzenia do bezpośredniego informowania ( pager ) i osobiste systemy alarmowe. Klientelę działu stanowią użytkownicy prywatni i przedsiębiorstwa o różnych profilach działalności. Dział Personal & business Communications pracuje głównie na rynku europejskim.

**Dział: Komunikacja radiowa ( Radiocom )**

jest kompetentny w zakresie telekomunikacji bezprzewodowej. Jego domeną są prywatne mobilne systemy łączności, systemy łączności komórkowej oraz systemy przeznaczone dla: policji, wojska i służb specjalnych. Działalność działu obejmuje rozwój, produkcję i dystrybucję urządzeń oraz systemów radiowych jak również budowę urządzeń łączności międzynarodowej. Szeroki zakres usług obejmuje także realizację systemów na zamówienia indywidualne.

**Dział: Automatyzacja usług**

wyspecjalizował się na rynku światowym w zakresie produktów, systemów i usług służących racjonalizacji i automatyzacji działalności usługowej. Nowoczesne urządzenia ułatwiają lub przejmują rutynowe czynności, dzięki czemu możliwe jest udoskonalenie jakości wykonywanych usług i rozszerzenie ich zakresu. Tacy oferenci usług jak np. banki czy przedsiębiorstwa komunikacji miejskiej stosując nowoczesną technikę zyskują istotną przewagę w stosunku do swych konkurentów i osiągają wyraźne korzyści ekonomiczne. Ten dział koncernu Ascom jest także wiodącym oferentem automatów do segregacji poczty.

**Dział: Dystrybucja i usługi**

Jego zadaniem jest wspieranie pozostałych działów w zakresie marketingu i dystrybucji. Zajmuje się on koordynacją działalności w poszczególnych krajach, co ułatwia rozwijanie aktywności zagranicznej. Dział korzysta z odpowiednich kanałów zbytu lub z własnej sieci dystrybucyjnej, zapewnia serwis i obsługę. Jest Partnerem dla innych działów i trzecich oferentów.

Temat referatu:

**sieci kalkulacyjne, rozdzielona obróbka danych**

Przedmiotem referatu jest prezentacja koncernu Ascom, a szczególnie jego działalności w dziedzinie sieci prywatnych jak również przedstawienie nowości technicznych i tendencji rozwojowych w zakresie tworzenia sieci łączących komputery i work stations. W dzisiejszym świecie rozdzielonej obróbki danych aspekty komunikacyjne ( np. sieci, miejsca podziału ) nabierają coraz większego znaczenia.







POLSKIE  
TOWARZYSTWO  
INFORMATYCZNE

digital

ascom

CasePol



IBM

