POLSKIE TOWARZYSTWO INFORMATYCZNE

CZTERNASTE JESIENNE SPOTKANIA PTI

Problemy jakości i niezawodności środków informatyki

Mrągowo, 16-20 listopada 1998r.

CZTERNASTE JESIENNE SPOTKANIA PTI

MRĄGOWO '98 – 16-20 LISTOPADA 1998

PROBLEMY JAKOŚCI I NIEZAWODNOŚCI ŚRODKÓW INFORMATYKI

XIV JESIENNE SPOTKANIA PTI - MRĄGOWO '98
16.11.98 - 20.11.98

Anna Cetnarowicz-Jutkiewicz

Narodowy Bank Polski

Warszawa

Szanowni Państwo

Trzydzieści trzy godziny wykładów, prezentacji i dyskusji to dwa semestry studiowania jednego przedmiotu. W dodatku, metodą „kilka w jednym" będą Państwo mieli okazję postudiować kilka przedmiotów na raz. Tegoroczne Mrągowo jest: sesją INSPIRE, seminarium poświęconym dokumentom, zajęciami z praktyki niezawodności systemów informatycznych, cyklem monograficznych wykładów na temat baz danych, grafiki i typografii komputerowej, wreszcie przeglądem platform systemowych oraz technik stosowanych w zarządzaniu przedsiębiorstwem.

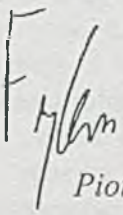
Układ konferencji jest próbą realizacji wszystkich postulatów, które zgłosili Uczestnicy oceniający poprzednie spotkania. Mamy nadzieję, że tegoroczni odbiorcy wspomogą naszą inwencję nowymi propozycjami.

Nasze spotkanie odbywa się kilkanaście dni przed drugim Kongresem Informatyki Polskiej. Na Kongresie będą znaczące postaci życia publicznego, wielu znajomych, z którymi warto się spotkać, będzie można podpisać się pod Paktem o Społeczeństwie Informacyjnym, każdy chętny będzie upoważniony do przedstawienia swoich opinii na temat naszych – informatyków – interesów. Mam nadzieję, że mrągowskie dyskusje oficjalne – a zwłaszcza kuluarowe – będą dobrym przygotowaniem do Kongresu. Tych, którzy się jeszcze nie zdecydowali poświecić dwóch dni bardziej aktywnym, niż samo słuchanie wykładów, formom uczestnictwa w życiu środowiska, gorąco do udziału w Kongresie zachęcam.

Jak zawsze w Mrągowie będziemy mówić o sprawach PTI. Rośnie nam ono w oczach od jesieni do jesieni: Oddział Wielkopolski właśnie przekroczył stu członków, reaktywował się Oddział Małopolski, coraz aktywniej działa koło w Gdańsku, rozwija się Europejskie Komputerowe Prawo Jazdy. Przyszłoroczna szkoła w Szczyrku będzie w części naukową, recenzowaną Pierwszą Krajową Konferencją Inżynierii Oprogramowania. Dwudziestego drugiego maja przyszłego roku zbierzemy się na siódmym w historii Towarzystwa Zjeździe. Nigdy jednak nie jest tak dobrze, żeby nie można było czegoś poprawić. Tych z Państwa, którzy przez niedopatrzenie nie są jeszcze członkami PTI namawiam do naprawienia tej pomyłki.

Pech, a tak naprawdę nasze błędy, sprawiły, że w trakcie naszego spotkania będzie się odbywać w Poznaniu SEES'98 (Software Engineering Education Symposium). Do tej konferencji PTI też przyłożyło rękę – w tym przypadku rękę Jurka Nawrockiego. Przepraszamy Jerzy, będzie nam Cię w Mrągowie brakować tak samo jak Tobie nas w Poznaniu.

Pozdrawiam z przedwczesnymi życzeniami świątecznymi. Jakoś trzeba będzie przetrwać do następnego listopada.



Piotr Fuglewicz

Tomasz Byzia

Narzędzia zapewnienia i kontroli jakości oprogramowania

Wstęp

Truizmem jest stwierdzenie, że dla użytkowników, dostawców oraz producentów, jakość systemów informatycznych jest rzeczą najważniejszą. Dla użytkownika wysoka jakość oznacza satysfakcję, rozwiązanie rzeczywistych problemów z przetwarzaniem dużej ilości danych, zaufanie do zastosowanych rozwiązań, niezawodność, bezpieczeństwo i „spokój" na długie lata. Dla dostawców wysoka jakość oznacza satysfakcję z dostarczonego systemu, uznanie w oczach klienta, niskie koszty konserwacji i wysokie zyski teraz i w przyszłości. Wysoka jakość jest hasłem, które tam gdzie się pojawia, przyciąga do siebie ludzi biznesu, bo wysoka jakość broni się sama i zawsze się opłaca.

Truizmem jest także stwierdzenie, że jakość dzisiejszych systemów informatycznych pozostawia wiele do życzenia. Mimo iż pierwsze objawy „kryzysu oprogramowania" zauważono już pod koniec lat 60-tych, niewiele udało się zrobić przez ostatnie 30 lat. Rozwój inżynierii oprogramowania pozwolił wypracować wiele metod, narzędzi, zaleceń, modeli i paradygmatów, które zbliżały nas do momentu w którym chciało się powiedzieć: „oto wiemy jak produkować niezawodne oprogramowania wysokiej jakości". Wystarczy jednak przejrzeć tytuły artykułów w kilku z współczesnych tygodników poświęconych zagadnieniom wytwarzania oprogramowania i informatyzacji, aby bez trudu odkryć, że dawne problemy są obecne i dzisiaj. Wpływa na to wiele czynników. Decydują o tym zarówno bardziej złożone architektury współczesnych systemów jak i wyższe wymagania klientów oczekujących wysokiej niezawodności, większej elastyczności i podatności na zmiany. Brakuje wdrożonych i skutecznych metodyk szacowania projektów, brakuje zasobów, czasu, pieniędzy, a jeśli nawet są to źle się nimi zarządza.

Żyjąc w takim świecie od dziesięcioleci, przyzwyczailiśmy się do tego, że przystępując do projektów informatycznych robimy to z pewnego rodzaju „podwójną moralnością". Teoria - teorią, a praktyka - praktyką. Każdy kierownik przedsięwzięcia podpisze się pod stwierdzeniem, że należy stosować zasady inżynierii oprogramowania, zapobiegać problemom, a nie „gasić pożary", kontrolować jakość, przeprowadzać przeglądy projektu, pracować dobrze, a nie źle, itd. Z drugiej strony dla większości zalecenia te pozostają czymś złudnym, niepraktycznym czy wręcz niewykonalnym. Stąd tylko krok do budowania teorii usprawiedliwiających własny brak konsekwencji. Metody są nieadekwatne, ludzie omylni, reguły niepraktyczne. W życiu osobistym zawsze spodziewamy się wysokiej jakości produktów i usług z których będziemy korzystać. W życiu zawodowym czasem zadowalamy się „bylejakością" i oczekujemy, że jakaś „magiczna siła" naprawi nasze błędy i poskleja rozpadające się przedsięwzięcie.

Poruszony przeze mnie tutaj problem jest jedynie wierzchołkiem góry lodowej. Jest skutkiem pewnego ogólnego i tajemniczego mechanizmu komplikowania sobie życia przez odrzucanie oczywistych zasad racjonalnego działania w miarę wzrostu złożoności i skomplikowania przedsięwzięć informatycznych. Jest to bardzo ważny i trudny problem znany zapewne każdemu. Nie chcę go jednak tłumaczyć i jednoznacznie wyjaśniać. Każdy musi jego znamiona odkryć w sobie samym

i sam znaleźć właściwe lekarstwo. Moja rola polega na dostarczeniu słuchaczom „materiału do przemyśleń". W tym celu przedstawiając szerokie spektrum narzędzi wspierających te procesy zapewnienia i kontroli jakości pragnę skłonić wszystkich do osobistej refleksji i uczciwej odpowiedzi na następujące pytania:

- czy wiedziałem o wymienionych narzędziach i metodach zapewnienia jakości?
- czy wiedziałem, że są takie proste i łatwe w stosowaniu?
- czy je stosowałem?
- dlaczego ich nie stosowałem?
- kiedy zacznę je stosować?

Jak powstaje jakość?

Jakość jest to według norm ISO 9000: „ogół cech i właściwości wyrobu/produktu lub usługi decydujących o jego/jej zdolności do zaspokojenia stwierdzonych lub przewidywanych potrzeb". Pożądane cechy wyrobu kształtowane są w cyklu wytwórczym, który nosi nazwę koła (spirali) jakości i obejmuje następujące procesy:

- marketing i badanie rynku
- tworzenie projektu/specyfikacji i konstruowanie wyrobu
- zaopatrzenie
- planowanie i rozwój procesów
- produkcja
- kontrola i badania jakościowe
- pakowanie i magazynowanie
- zbyt i dystrybucja
- instalowanie i uruchamianie
- pomoc techniczna, konserwacja i obsługa
- likwidacja

Jakość końcowa jest więc wypadkową jakości wymagań, jakości projektowania, jakości wykonania i jakości użytkowania. Działania te są ściśle związane z procesami transformacji idei i wymagań użytkownika na działający system informatyczny. Określają jakby horyzontalny wymiar procesów wytwórczych. Wymiar wertykalny związany jest natomiast z różnymi poziomami zarządzania procesami w przedsięwzięciu. Obejmuje wszystkie planowane i systematyczne działania, niezbędne do uzyskania wysokiego stopnia ufności, że wyrób lub usługa spełni ustalone wymagania jakościowe. Poziomy te tradycyjnie dzielimy na:

- zarządzanie jakością,
- zapewnienie jakości,
- kontrolę jakości.

Historycznie pierwsza była kontrola jakości. Sprowadza się ona do sprawdzania, mierzenia lub testowania jednej lub kilku charakterystyk produktu i odnoszenia wyników do wyspecyfikowanych wymagań w celu potwierdzenia zgodności. Zadanie to wykonywane zwykle przez wyspecjalizowany personel nie wchodzi w zakres obowiązków pracowników produkcyjnych. Produkty niezgodne ze specyfikacjami są odrzucane lub przekazywane do poprawienia. Klasycznym przykładem sposobu kontroli jakości jest testowanie oprogramowania.

Kolejny krok w procesie ewolucji systemów jakości zawdzięczamy klientom. Ruchy konsumenckie lat 70-tych sprawiły, że produkty musiały lepiej pasować do wymagań rynku - klienta. Zapewnienie jakości było naturalnym rozwinięciem poprzednich praktyk. Jakość zaczęto wbudowywać w produkt od początku przez prowadzenie systematycznych i zaplanowanych działań prowadzących do wytwarzania produktów zgodnych ze specyfikacją. W celu ciągłego zapewnienia jakości dokonywano regularnych inspekcji, przeglądów, audytów i zewnętrznych ocen. Rozbudowywano działy marketingu i analizy wymagań, odpowiedzialne za dostarczanie wiarygodnych i uzgodnionych z przyszłym użytkownikiem specyfikacji wymagań.

Najnowszą koncepcją, która nie neguje pozostałych lecz uzupełnia je o globalizację działań pro jakościowych, jest zarządzanie przez jakość. Jest to skoncentrowane na jakości zarządzanie przedsiębiorstwem, przy współudziale wszystkich jego członków, w którym długofalowe korzyści przedsiębior-

stwa oraz korzyści społeczeństwa osiągane są poprzez spełnienie oczekiwań klientów.

Przegląd narzędzi

Na każdym poziomie związanym z zarządzaniem jakością dysponujemy określonym zestawem narzędzi, metod i technik, które pozwalają na sprawne i efektywne realizowanie zadań. Narzędzia są tutaj rozumiane bardzo szeroko jako wszelkie środki pomocne w osiąganiu celów.

W dalszej części zostaną omówione wybrane narzędzia wraz z krótką charakterystyką celu i zakresu ich stosowania.

Narzędzia zarządzania jakością

- Polityka jakości
- Metryki jakości
- Audyt systemu jakości

Narzędzia zapewnienia jakości

- Praca zespołowa
- Inżynieria wymagań
- Metodyka projektowania
- Weryfikacja i walidacja
- Zarządzanie zmianami

Narzędzia kontroli jakości

- Diagramy procesów
- Histogramy
- Diagramy Pareto
- Wykresy korelacji
- Karty kontrolne
- Listy kontrolne
- Diagramy Ishikawy

Podsumowanie

Przedstawione narzędzia są bardzo proste i łatwe w użyciu. Opanowanie każdego z nich zajmuje najwyżej 1 godzinę. Mimo swojej prostoty dostarczają bogatych możliwości oceny, weryfikacji i kształtowania jakości produktów. Ich konsekwentne stosowanie tam, gdzie rzeczywiście ma miejsce przynosi znaczne korzyści. Wracamy jednak tutaj do problemu postawionego na początku: dlaczego takie proste i „tanie" narzędzia nie są w praktyce stosowane. Czy są za proste, wręcz trywialne, czasochłonne, pracochłonne, nieefektywne, nieefektowne? Zapraszam do dyskusji.

Piotr Dembiński

Instytut Podstaw Informatyki

Polskiej Akademii Nauk

ul. Ordona 21, Warszawa

e-mail: piotrd@ipipan.waw.pl

*Metody formalne w projektowaniu oprogramowania: mity i rzeczywistość**

1. Wstęp

Metody formalne obecne są w informatyce od jej początku, ale termin ten w inżynierii oprogramowania pojawia się dopiero, gdy jego burzliwy rozwój i rosnąca złożoność wywołały wyraźną potrzebę bardziej uporządkowanego i zorganizowanego podejścia do całego procesu produkcji softwaru. Już w latach sześćdziesiątych podjęto próby wskazania reguł i konstrukcji, które programista (zespół programistów) powinien stosować, a które omijać. Pojawiły się propozycje metod znanych pod chwytliwymi nazwami, jak: programowanie strukturalne (*structured programming*), systematyczne (rygorystyczne) programowanie (*systematic (rigorous) programming*), projektowanie poprzez stopniowe przekształcania opisu problemu do jego rozwiązania programistycznego (*stepwise program refinement, top-down design*) itp. Metody te próbowano coraz bardziej formalizować, łącząc naturalną konieczność pewnej systematyzacji inżynierskiego postępowania z pokusą wykorzystania rosnących możliwości komputerów do automatyzacji tego postępowania.

Akademickie rozważania na temat formalnych specyfikacji i weryfikacji zadań programistycznych zderzyły się z rzeczywistością końca lat siedemdziesiątych, które zwykło się uważać za lata postępującego kryzysu w dziedzinie projektowania softwaru. Właśnie wtedy zauważono ogromnie szybko wzrastające koszty oprogramowania (przy taniejącym ciągle sprzęcie) szczególnie odczuwalne w zestawieniu z jego rosnącą zawodnością. Niektóre z wielkich projektów nie zostały ukończone, mnożyły się spektakularne przykłady błędów, które wymagały nadmiernych nakładów na ich poprawianie, a to z kolei często wprowadzało nowe usterki, nad którymi, po pewnym czasie, nikt nie panował. Szybko obliczono, że w cyklu życia oprogramowania najwięcej kosztują błędy powstające w początkowej fazie projektowania. Dlatego właśnie zainteresowano się bardzo poważnie metodami formalnymi. Wydawało się niektórym, że ich zastosowanie w praktyce może być remedium na powstałą, kryzysową

* Niniejsza praca stanowi zmodyfikowaną wersję wykładu wygłoszonego na plenarnej sesji Krajowego Sympozjum Telekomunikacji, które odbywało się w dniach 9-11 września 1998 r. w Bydgoszczy..

sytuację. Te nadzieje były, niestety, podtrzymywane przez wiele nieodpowiedzialnych obietnic, które tworzyły niebezpieczny mit nie mający pokrycia ani w rzeczywistych możliwościach metod formalnych, ani w możliwościach dostępnej wtedy technologii komputerowej tam, gdzie metody te mogłyby mieć zastosowanie. Po kilku latach nastąpiło zniechęcenie, które utrwaliło poglądy zdecydowanie sceptyczne (nawet jeśli głośno nie wyrażane), co do praktycznej przydatności metod formalnych. Powstał zatem drugi niebezpieczny mit głoszący, że metodami tymi nie warto się poważnie zajmować.

W poniższym szkicu spróbujemy przekonać czytelnika, że odpowiednio rozważnie rozumiane metody formalne zawsze miały i mają swoje własne miejsce w całym złożonym procesie wytwarzania oprogramowania. Wskażemy na konieczne warunki, które muszą zaistnieć, żeby ich użyteczność była dostrzeżona, a w szczególności zwrócimy uwagę na potrzebę istnienia efektywnych narzędzi automatyzujących wiele żmudnych czynności w systematycznym projektowaniu. W tym kontekście pokażemy też, jak postęp technologiczny umożliwia realizację niektórych pomysłów, których (pozorna) prostota i spektakularność otrzymywanych wyników może pomóc w przełamywaniu obiegowych poglądów. Zwrócimy też uwagę na to, że metody „specjalizowane", ograniczone do danej dziedziny zastosowań wydają się spełniać łatwiej oczekiwania odbiorców. Tę tezę zilustrujemy ogólną charakterystyką metod formalnych i stosowanych narzędzi w projektowaniu protokołów komunikacyjnych oraz wnioskami z realizacji projektu europejskiego, którego jednym z zadań było wspomagana komputerowo specyfikacja i implementacja protokołu XTP (*EXpress Transport Protocol*).

2. Inżynieria oprogramowania i jej formalne metody

Inżynieria oprogramowania operuje terminem cyklu życia systemu informatycznego, obejmującego jego różne etapy (fazy). Liczba tych etapów, ich nazwy i powiązania różnią się w zależności od autora, ale zasadniczo panuje zgoda, że cykl taki zawiera interesujące nas tutaj etapy:

- specyfikacji,
- projektowania,
- implementacji.

Każdy z nich można podzielić na podetapy. Zakłada się zwykle, że wyniki uzyskane w kolejnym (pod)etapie mogą spowodować konieczność powrotu do etapów poprzednich. W tym sensie, na przykład, uzyskujemy w kolejnych iteracjach cyklu prototypową, a nie ostateczną implementację.

Przejście z etapu na etap może być mniej lub bardziej sformalizowane i w odpowiednim zakresie wspomagane przez specjalistyczne narzędzia programistyczne. Na każdym etapie pracujemy z jakąś formą opisu projektowanego systemu, przy jednym założeniu niezmiennym: w wyniku fazy implementacji otrzymany opis jest gotowy do wykonania przez komputer lub szerzej - w środowisku komputerowym, np. w sieci. Znaczy to, że jego kształt jest całkowicie sformalizowany. Jeżeli taki sformalizowany opis otrzymujemy automatycznie z opisu powstałego na wcześniejszym etapie, to warto pamiętać, że również ten wcześniejszy opis musi być całkowicie formalny. Tak więc stopień automatyzacji procesu wytwarzania systemu informatycznego (oprogramowania) w bezpośredni sposób zależy od stopnia jego sformalizowania. W tym sensie powiedzieć możemy, że metody formalne towarzyszą informatyce (inżynierii oprogramowania) od jej początku, chociaż zakres ich zainteresowań zmieniał się, w miarę jak uzyskiwano zadowalającą automatyzację pewnych etapów projektowania.

W bardzo wczesnym etapie rozwoju informatyki problemy inżynierii oprogramowania w tym schematycznym ujęciu koncentrowały się wokół relacji pomiędzy:

- programami w językach programowania,
- programami asemblerowymi,
- bezpośrednim kodem komputerowym (kto dziś odróżnia te dwie ostatnie kategorie!).

Przez lata zajmowano się składnią i semantyką języków formalnych w poszukiwaniu efektywnych metod automatyzacji przejścia od opisu programu (w językach takich jak Algol, PL1, Pascal, Algol 68 i wielu dziesiątkach innych) do jego kodu komputerowego. Te często bardzo matematycznie wyrafinowane rozważania zaowocowały tym, że dzisiaj istnieją efektywne kompilatory, a języki programowania, takie jak C, spełniają rolę asemblerów z przeszłości.

Z biegiem lat programowanie zmieniło swój charakter, ponieważ skala i stopień złożoności problemów, które rozwiązuje się za pomocą komputerów, wymagają zaangażowania i współdziałania wielu ludzi, bogatego i często rozproszonego terytorialnie środowiska obliczeniowego oraz wyrafinowanych narzędzi, wspomagających cały proces projektowania, uruchamiania i utrzymywania systemów informatycznych. Jednym z przejawów tych zmian, a także wyników wcześniejszych prób jest fakt, że w schemacie, którym posługujemy się powyżej, uwaga skoncentrowana jest na etapach poprzedzających kodowanie w wybranym języku programowania. Okazuje się bowiem, że wybory dokonane w tych fazach są decydujące dla powodzenia dużych przedsięwzięć programistycznych. W tym sensie niesłychanie istotna jest faza sformułowania wymagań projektowych (*requirement specification*), a następnie przełożenie tych wymagań na specyfikację systemową (*system level specification*), tzn. taką, która w znacznie większej mierze zajmuje się "logiką" i "architekturą" proponowanego rozwiązania niż oprogramowaniem poszczególnych funkcji i ich efektywnością. Tak więc współczesna inżynieria programowania, w schematycznym ujęciu fragmentu jej zainteresowań, który przedstawiliśmy powyżej, koncentruje się wokół relacji pomiędzy:

- specyfikacją wymagań,
- specyfikacją systemową,
- programami w językach programowania.

Stworzenie metod i narzędzi wspomagających proces wytwarzania systemów informatycznych na tych etapach - wymaga precyzyjnego określenia, czego oczekujemy i jak rozumiemy specyfikację wymagań, a jak specyfikację na poziomie systemowym. Nieuchronnie więc mówimy o formalizacji i metodach formalnych, co najmniej w stopniu i części, w której zakładamy pomoce automatyzujące w pewnym zakresie konieczną do wykonania pracę.

W tym ogólnym wstępie nie będziemy nawet próbowali dokonywać przeglądu proponowanych formalizmów, opartych o nie metod i związanych z nimi narzędzi. Odsyłamy zainteresowanego czytelnika np. do [5] lub popularnego artykułu [11] i podanych tam referencji. Bardziej istotne jest wyjaśnienie, czego po tych metodach i narzędziach oczekujemy i na jakie trudności w ich stosowaniu napotykamy.

Specyfikacja wymagań, to zbiór tego, czego spodziewamy się po projektowanym systemie, jakie funkcje ma on spełnić i jakie wymierne efekty przynieść. Jest to więc, z grubsza rzecz biorąc, określenie problemu, który chcemy rozwiązać, poprzez zdefiniowanie jego najważniejszych atrybutów. Rozwiązań w takim sformułowaniu może nie być wcale (tzn. problem jest źle sformułowany), może ich być kilka istotnie różnych, a nawet nieskończenie wiele. Ważne jest, żeby sformułowanie wymagań nie ograniczało nadmiernie możliwości wyboru rozwiązania, ale też by wykluczało rozwiązania, w sposób oczywisty, nierozsądne. Jak się okazuje, jest to trudne w potocznym rozumieniu, czego dowodzą liczne przypadki nieporozumień pomiędzy odbiorcami, a projektantami systemów (nie tylko) informatycznych. Sformalizowanie wymagań - to problem znacznie trudniejszy.

Specyfikacja systemowa to, jak powiedzieliśmy wyżej, określenie zasadniczej koncepcji rozwiązania problemu, zadanego poprzez specyfikacje wymagań. Opis rozwiązania zależy w ogromnym stopniu od języka, którym się posługujemy. Powiedzmy więc tylko, że taki opis chcielibyśmy traktować poniekąd, jak program wysokiego poziomu, w którym – w pierwszym przybliżeniu - nie zwracamy uwagi na szczegóły i rygory realizacyjne, natomiast przywiązujemy szczególną wagę do wypełnienia wszystkich wymagań. Podejmujemy także zasadnicze decyzje ustalające:

strukturę i hierarchie elementów projektowanego systemu, rodzaj komunikacji pomiędzy nimi itp.

Program (programy) w języku programowania (takim, jak np. C lub C++), to konkretna implementacja specyfikacji systemowej, biorąca pod uwagę docelowy system komputerowy i inne wymogi realizacyjne. Mamy więc tutaj do czynienia ze zbiorem wymagań implementacyjnych, np. dotyczących ograniczeń czasowych, różnych od tych z poziomu specyfikacji problemu. Niespełnienie tych wymagań przez implementację może powodować konieczność zmiany specyfikacji systemowej, a nawet niekiedy i wstępnych wymagań, dotyczących definicji problemu. Stąd iteracja etapów projektowania.

Mając określone "produkty" poszczególnych etapów, łatwo jest przewidzieć to, w czym oczekujemy pomocy i co chcielibyśmy automatyzować. O ile trudno wyobrazić sobie, że sformułowanie wymagań mogłoby automatycznie generować specyfikację systemową - konieczne są tutaj zasadnicze wybory i koncepcje, które nie mogą się obyć bez interwencji projektanta - to przejście od specyfikacji systemowej do konkretnej realizacji programistycznej wydaje się krokiem bliskim takiej (wspomaganej i, być może, niekompletnej) automatyzacji.

Podsumowując, metody i narzędzia, których oczekujemy to:

- bogate, a jednocześnie łatwo czytelne i "przyjazne", formalizmy (techniki) definiowania specyfikacji wymagań i specyfikacji systemowych oraz narzędzia wspomagające formułowanie poprawnych i spójnych specyfikacji (edytory tekstowe i graficzne, analizatory składni i semantyki, systemy wspomagające uzyskiwanie i dokumentację kolejnych wersji specyfikacji itp.);
- metody i narzędzia wspomagające weryfikację wymagań (sformułowanych na poziomie specyfikacji wymagań) przez specyfikacje systemowe;
- metody i narzędzia generowania kodu w docelowym języku programowania bezpośrednio z zadanej specyfikacji systemowej;
- metody i narzędzia wspomagające instalację, optymalizację i testowanie otrzymanego kodu w konkretnej konfiguracji komputerowej i programowej (system operacyjny, oprogramowanie komunikacyjne itp.). W dużym zakresie tego rodzaju narzędzia istnieją i są związane z konkretnym językiem programowania (kompilatory, debuggery, itp.) - w fazie uruchamiania potrzebne są jednak na ogół dodatkowe zabiegi (często wymagające interwencji człowieka, który system uruchamia) i specjalizowane biblioteki procedur potrafiące skompilowane części specyfikacji systemowej połączyć w jedną wykonywalną całość, włączoną do istniejącego środowiska.

Jeżeli mówimy o skomplikowanym oprogramowaniu systemowym (protokole komunikacyjnym, poważnej aplikacji itd.), to dodatkowe zabiegi - o których mowa w ostatnim punkcie powyżej - oraz w miarę kompletne testowanie w rzeczywistym środowisku, mogą poważnie wydłużyć i podrożyć proces uzyskania zadowalającej implementacji. Spełnienie wymagań implementacyjnych, np. wydajnościowych, jest, na ogół, bardzo trudne. Wymaga często wielokrotnego powrotu i zmian we wcześniejszych fazach projektowania. Wydaje się więc zasadne dążenie do skrócenia każdego cyklu, poprzez możliwość uzyskania estymacji parametrów implementacyjnych jeszcze przed właściwą implementacją (wykonywalną instalacją) oprogramowania. Myśl to nie nowa, a osiągnięcie tego celu można uzyskać modelując (specyfikując) projektowany system razem ze środowiskiem, w którym przewidywane jest jego wykonanie, a następnie analizując wymagania implementacyjne bezpośrednio w tym modelu za pomocą środków analitycznych lub symulacyjnych. Jeżeli przyjmiemy takie rozwiązanie za istotny element wyposażenia projektanta, to do powyżej wymienionych punktów dodamy:

- metody i narzędzia analizy wymagań implementacyjnych, np. czasowych i wydajnościowych, na poziomie specyfikacji systemowych, lub na poziomie pośrednim pomiędzy taką specyfikacją, a implementacją.

Współczesna inżynieria oprogramowania próbuje sprostać oczekiwaniom krótko scharakteryzowanym powyżej. Istnieje wiele modeli, formalizmów, metod i narzędzi. Technologiczny postęp, jaki dokonał się w ostatnim dziesięcioleciu, pozwala realizować zamysły, o których nie można było marzyć jeszcze niedawno. Przykładem na to może być postęp jaki dokonał się w zastosowaniu tzw. metody sprawdzania (w) modelu (*model checking*), która służy automatycznej weryfikacji własności projektowanego systemu zadanych w postaci pewnych formuł logicznych. Metoda ta, do pewnego stopnia, spełnia naiwne marzenie, które można wysłowić następująco:

- "gdyby istniały niesłychanie szybkie i pojemne (w sensie pamięci) komputery, to dla każdego systemu (programu) można by sprawdzić zadaną własność, przeglądając jego pełny graf stanów".

Taka rzeczywiście jest zasadnicza myśl tej metody. Modelami, o których tutaj mowa, są zbiory (lub drzewa) obliczeń definiowane przez taki graf stanów. Skończoność zbioru stanów daje się (częściowo) uzasadnić, jeśli pamiętamy, że zbiór wartości, które możemy reprezentować w komputerze, jest zawsze ograniczony. Łatwo jednak zauważyć, że liczba stanów rośnie bardzo szybko (wykładniczo) w stosunku do rozmiaru systemu, szczególnie jeśli mówimy o modelach dla systemów rozproszonych i współbieżnych. Czy więc metoda jest zupełnie nierealna? Okazuje się, że nie. Stosując wyrafinowane techniki reprezentacji danych i redukcji przestrzeni stanów, w połączeniu z niewiarygodnym postępem technologicznym, obserwowanym ostatnio, okazuje się, że można mieć nadzieję na praktyczne stosowanie tej metody. Dość powiedzieć, że udaje się weryfikować, w rozsądnym czasie, systemy o liczbie stanów sięgającej 10^{20} (zob. np. [2]). Dla metody weryfikacji w modelu naturalnymi przykładami testującymi były układy logiczne i proste protokoły komunikacyjne. Własności w postaci formuł logicznych (temporalnych, czy ogólniej modalnych) stosunkowo dobrze pasują do definicji wymagań w takich przykładach.

Nie będziemy tego tematu tutaj rozwijać i zainteresowanego czytelnika odsyłamy np. do książki [7], gdzie znajdzie potrzebne referencje i przykłady dotyczące protokołów. Takie przykłady mogą napawać pewnym optymizmem, chociaż metody formalne inżynierii oprogramowania dalekie są od powszechnego stosowania.

3. Metody formalne w projektowaniu protokołów komunikacyjnych

W krótkiej charakterystyce oczekiwanych metod i narzędzi, wspomagających wytwarzanie oprogramowania na początkowych jego etapach, skupiono uwagę na tych, które mają charakter uniwersalny, niezależny od dziedziny zastosowań. Jak słusznie zauważają różni autorzy (np. P. Zave w [4]), wydaje się dzisiaj zupełnie niemożliwe stworzenie metod i narzędzi równie pomocnych w produkcji oprogramowania we wszystkich dziedzinach. Ogólne zasady, powszechnie słuszne, nie zastąpią wiedzy i doświadczenia inżynierskiego, związanego z wybranymi zastosowaniami. Projektowanie protokołów potwierdza, naszym zdaniem, ten pogląd. Wiele czynników miało wpływ na to, że metody formalne dość wcześnie pojawiły się w projektowaniu protokołów. Nie bez znaczenia jest tutaj fakt, że ta dziedzina wymaga, jak mało która, kodyfikacji i normalizacji. Nie do przecenienia w tym kontekście jest rola bazowego modelu (*Basic Reference Model*) OSI (*Open Systems Interconnection*) [10], nad którym pracę w ISO (*International Standards Organization*) rozpoczęto w połowie lat siedemdziesiątych, a ostatecznie upowszechniono w formie normy w roku 1979. Za tym modelem nastąpiło szybkie normalizowanie protokołów różnego poziomu.

Wysiłek normalizacyjny spowodował, że w gronie specjalistów wytworzył się w miarę wspólny zestaw pojęć, terminów, notacji i sposobów na wyrażanie podobnych własności. W standardach komunikacyjnych stosuje się zestaw tych znanych środków, dobrze (na ogół) dostosowanych do sytuacji, co powoduje, że zapis norm, w dużej części przecież nie formalny, stanowi dobry punkt odniesienia z chwilą przejścia do definicji sformalizowanych. Model OSI ma tę

właściwość, że sytuuje i definiuje protokół komunikacyjny - centralny obiekt swego zainteresowania - w kontekście otoczenia (warstwowego środowiska), w którym on będzie działał i jednocześnie określa jego wyabstrahowaną rolę w tym kontekście, a mianowicie usługę, którą protokół danej warstwy, wraz z kontekstem niższych warstw, oferuje użytkownikowi warstwy wyższej. Mamy więc w tym przypadku sytuację (z punktu widzenia metod inżynierii oprogramowania, w tym metod formalnych) koncepcyjnie niesłychanie klarowną: specyfikacja usługi protokolarnej pełni rolę specyfikacji wymagań, a sam protokół, sprzężony z niższymi warstwami, ma stanowić rozwiązanie spełniające te wymagania. Dodatkowym ułatwieniem na tym poziomie koncepcyjnym jest fakt, że protokoły skupiają uwagę na samej komunikacji (i jej jakości), nie wnikając w treść przesyłanej informacji. Obrazowo rzecz ujmując, w przesyłanych wiadomościach pomiędzy użytkownikami i w interfejsach międzywarstwowych interesuje nas, jak postępować, sprawdzać i uzupełniać informacje, podawane na kopercie listu, a nie to, co jest zawartością koperty. Wymagania, dotyczące treści i formy informacji na kopercie, mogą być skodyfikowane i znormalizowane, ponieważ ich interpretacja nie zależy od poszczególnego użytkownika. Protokół, a także usługa protokolarna, określone zostają w zasadzie poprzez wymianę tak znormalizowanych komunikatów.

Opisy protokołów i ich usług – znormalizowanych lub nie - w języku naturalnym (angielskim), stanowią podstawę specyfikacji wymagań. Można powiedzieć, że nic nie może ich całkowicie zastąpić na tym poziomie i że zawsze takie opisy będą dla człowieka (a więc projektanta) punktem odniesienia i źródłem rozumienia jakichkolwiek formalizmów. Stosuje się tutaj także pewne specyficzne notacje, tablice i diagramy stanów i przejść oraz przepływu wiadomości w czasie MSC (*Message Sequence Charts*) [6]. Całkowita formalizacja wymagań potrzebna jest wtedy, gdy mamy możliwość ich weryfikacji (najchętniej automatycznie - np. poprzez sprawdzanie modelu, o którym wspomnieliśmy wcześniej) w tworzonym oprogramowaniu. Próby takiej formalizacji są czynione i najczęściej wykorzystuje się tutaj języki dla różnych odmian logiki temporalnej, tzn. takiej, w której można wyrazić własności wiążące zdarzenia obliczeń w ich przeszłości, teraźniejszości i przyszłości. Trzeba od razu jednak zaznaczyć, że formalizmy logik temporalnych tylko w pewnym stopniu wyrazić mogą ograniczenia czasowe, które chcielibyśmy móc reprezentować. Jednakże doświadczenia zdobyte przy wykorzystaniu np. systemu weryfikacji SPIN [2] są wielce obiecujące.

Prace normalizacyjne i pojawiające się przy nich trudności sprawiły, że stosunkowo wcześnie powstał w ISO pomysł zaprojektowania specjalnych języków dla formułowania standardów usług i protokołów - języków wolnych od niejednoznaczności i niejednorodności, dotychczas używanej mieszaniny języka naturalnego z tworzonymi ad hoc specjalnymi notacjami. Około 1980 roku powstała w ISO, w której podjęto prace nad dwoma takimi językami, reprezentującymi dwa różne podejścia do problemu specyfikacji: jednym (Estelle) opartym na rozszerzonym pojęciu (komunikujących się) automatów i drugim (Lotos), którego podstawę stanowi algebra (komunikujących się) procesów. Wcześniej znany był język SDL – znormalizowany przez CITT - którego model miał tę samą podstawę, co Estelle. Wszystkie trzy razem występują często pod wspólną nazwą FDTs (*Formal Description Techniques*) [13], chociaż ostatnio ta nazwa staje się generyczną dla wielu innych, nie znormalizowanych formalizmów.

W końcu lat osiemdziesiątych zakończono prace normalizacyjne nad tymi trzema językami (ostatnia wersja SDL w roku 1992). Równolegle, wokół tych języków, powstawały mniej lub bardziej prototypowe narzędzia, ułatwiające ich używanie. Okazało się jednak, że przypisywany początkowo cel powstania FDTs polegający na chęci stopniowego eliminowania języka naturalnego z opisów norm, nie da się łatwo zrealizować, albo też nie jest to w istocie cel podstawowy. W coraz większym stopniu eksperymentowano z tymi językami, jak niegdyś z językami programowania wysokiego poziomu. Traktowano je jako takie, na które wygodnie jest tłumaczyć

nieformalne zapisy norm i jednocześnie wykorzystać, jako podstawę do otrzymania prototypowych implementacji, których działanie można sprawdzić w środowisku rzeczywistym lub symulacyjnym. Stąd też postulowane przez nas usytuowanie tej klasy języków w kategorii specyfikacji systemowej dla odróżnienia zarówno od języków specyfikacji wymagań, jak też od powszechnie używanych języków programowania.

Jak powiedzieliśmy uprzednio, języki poziomu systemowego tylko wtedy będą praktycznie stosowane jeżeli dają się przekształcić (skompilować) w wyrażenia języka programowania, takiego jak C. Taka kompilacja jest potrzebna nie tylko, gdy chcemy protokół rzeczywiście uruchomić w węźle sieci, ale także gdy chcemy go wykonywać, poprzez symulację w środowisku modelowanym. Jednocześnie, nie można wymagać, by taka kompilacja prowadziła do efektywnych implementacji, ponieważ mechanizmy definicyjne na poziomie specyfikacji systemu nastawione są raczej na modelowanie, a nie wykonanie. Przy założeniu, że kompilator działa poprawnie, otrzymujemy jednak – w dużej części automatycznie - działający poprawnie prototyp protokołu.

Na ogół, o efektywności implementacji przekonać się możemy dopiero po zainstalowaniu otrzymanego oprogramowania w konkretnym środowisku komputerowym (sieciowym). Co można zrobić, by przewidywać jego efektywność wcześniej, na etapie projektowania? Klasyczną metodą w takim przypadku jest stworzenie modelu kolejkowego (*queuing networks*) dla środowiska implementacyjnego, zawierającego projektowany moduł (np. protokół) i poddanie go analizie istniejącymi na rynku narzędziami, specjalnie dostosowanymi do tego modelu. To podejście ma swoje zalety, ponieważ model kolejkowy i stworzone dla niego metody analizy mają ugruntowaną pozycję. Ma jednak i tę wadę, że nie wykorzystuje modelu, w którym powstała specyfikacja systemowa. Zmusza więc do pracy z dwoma modelami, których wzajemną relację ustala, na ogół, intuicja i doświadczenie projektanta(ów). Istnieją więc próby by istniejące narzędzia pracujące bezpośrednio na specyfikacjach w konkretnym języku (Estelle, SDL, Lotos itp.), takie jak symulator, uzupełnić specjalnymi funkcjami, na wzór tych przystosowanych do modelu kolejkowego, które pozwalają na różne oceny wydajnościowe symulowanego systemu. Takie narzędzia zaprojektowano dla Estelle (EDT) [1] i w pewnym stopniu dla SDL (SDT) (zob. [6]). W tym pierwszym przypadku stworzone były i następnie używane w projekcie COP 62, o którym piszemy więcej poniżej.

Proces projektowania i implementacji protokołów zawiera specyficzne zadania, których rozwiązanie wymaga zarówno formalizacji, jak i automatyzacji. Dwa takie specyficzne zadania charakteryzujemy poniżej.

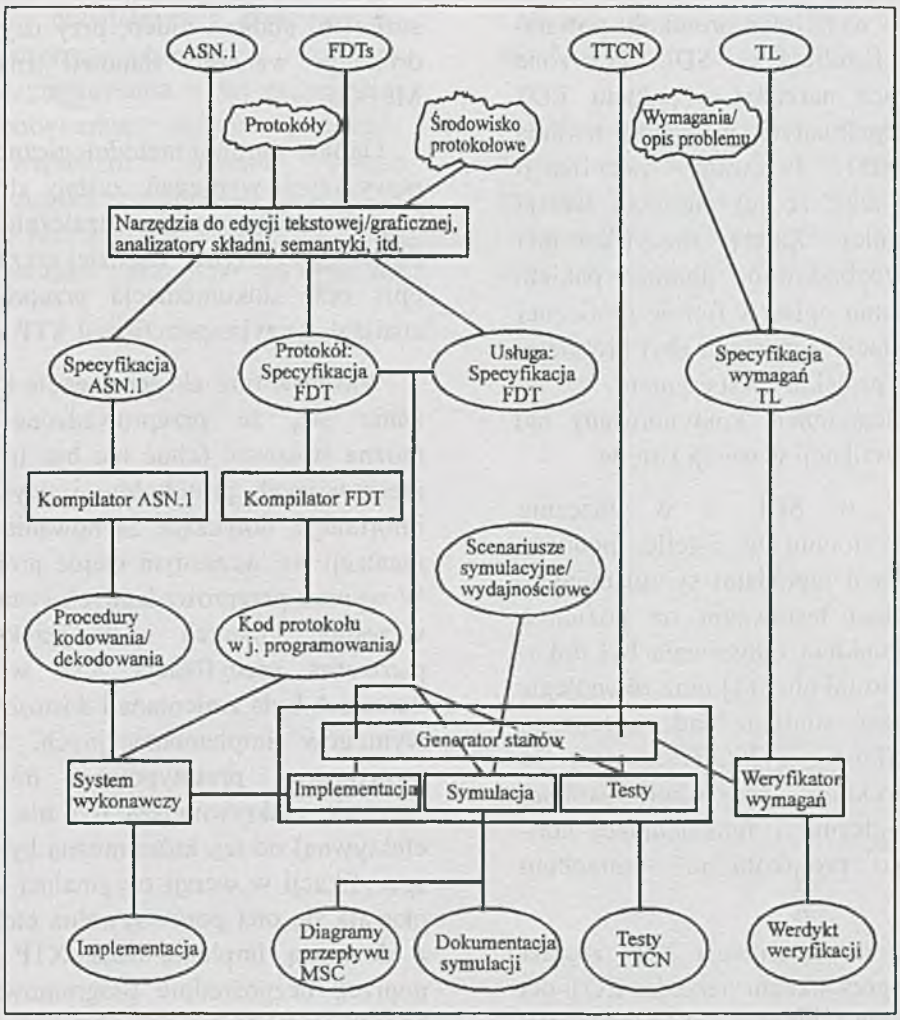
Protokoły komunikacyjne służą wymianie informacji zrozumiałej dla obu stron nawet wtedy, gdy mówią one - stosując terminologię potoczną - innymi językami. Aby to osiągnąć, trzeba pewnej umowy, precyzującej sposób, w którym informacje są zapisywane i jednoznacznego systemu ich kodowania (dekodowania) w ciągi bitów, które ostatecznie przesyłane będą poprzez łącza fizyczne. Te wymagania wymuszają ustalenie ogólnie przyjmowanego standardu obu tych elementów: zapisu na poziomie struktury danych (*abstract syntax*) i na poziomie kodu binarnego (*concrete syntax*). Elementem łączącym te zapisy jest jednoznaczny system kodujący. Istnieje taki standard dla definiowania struktur danych, tzn. język ASN.1 (*Abstract Syntax Notation One* - cyfra na końcu ma wyrażać możliwość powstawania różnych, znormalizowanych notacji) [8] i sposób kodowania jego wyrażeń (*Basic Encoding Rules*) [9]. Jeżeli myślimy o pewnym środowisku wspomagającym projektowanie protokołów, to nie może zabraknąć narzędzi właściwego kodowania i dekodowania. Istnieją próby integracji takich narzędzi z narzędziami związanymi z implementacją języków specyfikacji systemowej.

Inną specyfiką definicji protokołów i usług telekomunikacyjnych jest dobre rozgraniczenie pomiędzy tym, co jest w każdej implementacji danego protokołu konieczne, a tym, co pozostawione jest do decyzji projektanta konkretnej jego wersji (tzw. profilu). Testowanie protokołu, na poziomie

jego specyfikacji systemowej, dotyczy sprawdzenia semantycznej poprawności przekształcenia normy, zapisanej nieformalnie w wyrażenie sformalizowane. Na poziomie implementacji mamy do czynienia z testami, które wiążą się z normalną praktyką programistyczną i wymagają sprawdzenia jakości kodu wyprodukowanego przez kompilator oraz tymi (*conformance tests*), które sprawdzają czy implementacja jest zgodna z deklarowanym profilem. Z naszego punktu widzenia ważne jest, że zaproponowano sformalizowaną notację TTCN (*Tree and Tabular Combined Notation*), służącą opisowi testów tego rodzaju i określono wymagania na ich powstawanie i stosowanie. Istnieją także poważne próby generacji testów w notacji TTCN bezpośrednio ze specyfikacji protokołu w wyżej wymienionych językach poziomu systemowego. Podobnie jak w poprzednim przypadku, stwarzając środowisko projektowania protokołów komunikacyjnych nie można pominąć zadania automatyzacji powstawania, reprezentacji i generacji testów zgodności (zob. np.[12]).

Wymienić można jeszcze inne specyficzne cechy i wymagania na specjalistyczne oprogramowanie w interesującej nas dziedzinie. Wydaje się jednak, że krótka charakterystyka, którą przeprowadziliśmy, przekonuje do tego, że poprzez ograniczenie się do projektowania protokołów komunikacyjnych otrzymujemy istotne informacje szczegółowe, które pozwalają mieć nadzieje, że metody formalne komputerowo wspomagane, mogą dać efekty, na które trudno jest liczyć, jeżeli mówimy o oprogramowaniu w ogóle, nie odnosząc się do specyficznej dziedziny zastosowań. Trzeba jednak jasno powiedzieć, że odległa jest jeszcze perspektywa stosowania takiego systematycznego podejścia w powszechnej praktyce.

Na zakończenie tego paragrafu pokazujemy na Rys. 3.1 schemat hipotetycznego, zintegrowanego systemu, wspomagającego projektowanie i implementację protokołów komunikacyjnych, który byłby niejako próbą połączenia formalizmów, metod i narzędzi, opisanych krótko powyżej. Ze znanych autorowi systemów jedynie kilka zawiera zaledwie fragmenty tego schematu o znacznym stopniu zintegrowania. Są to przede wszystkim narzędzia wokół języka SDL, np. SDT oferowany przez Telelogic ze Szwecji i francuski GEODE firmy Verilog (o narzędziach SDL zob. np. [6]) a także, w pewnym stopniu, EDT (*Estelle Development Toolset*) [1]. W wielu innych systemach rozwinięte są tylko pojedyncze elementy prezentowanego na rysunku schematu.



Rys. 1. Schemat zintegrowanego systemu projektowania protokołów.

4. Przykład zastosowania

Chcielibyśmy zakończyć te z natury rzeczy ogólne uwagi krótkim opisem etapów i wyników zakończonego już projektu, mającego na celu otrzymanie prototypowej implementacji protokołu XTP (*EXpress Transfer Protocol*), w wersji 4.0 [14], bezpośrednio i (prawie) automatycznie otrzymanej z jego formalnej specyfikacji systemowej, również stworzonej w ramach tego projektu. Mówimy tutaj o projekcie Komisji Europejskiej Copernicus 62 pod nazwą "*High Speed Communication Systems Supporting Multimedia Applications - Selected Issues*" i realizowanym przez sześciu partnerów z czterech krajów (IT PW, IPI PAN i Alcatel z Polski, INT z Francji, PUB z Rumunii i UTC ze Słowacji). Projekt trwał dwa lata i zakończył się w kwietniu 1997 roku. Podajemy tutaj odwołanie tylko do raportu końcowego [3] tego projektu, w którym można znaleźć dokładny opis jego wyników, listę raportów i ogólnie dostępnych publikacji na jego temat.

Pierwszym z zadań projektu było zainstalowanie na Politechnice Warszawskiej i Politechnice w Żilinie (Słowacja) szybkiej sieci typu FDDI, jako rzeczywistego środowiska, w którym przeprowadzono by końcowe eksperymenty z przesyłaniem informacji multimedialnych pomiędzy węzłami sieci z zainstalowaną w nich prototypową implementacją XTP, o której była mowa powyżej.

Specyfikacje wyjściowe protokołu powstały zarówno w Estelle jak i SDL. Tworzone były za pomocą narzędzi z pakietu EDT (częściowo uzupełnianych w trakcie trwania projektu) i SDT. Przewagą specyfikacji w Estelle był fakt, że jej pierwsza wersja istniała wcześniej. Zaletą specyfikowania w SDL były rozbudowane pomoce pakietu SDT dla tworzenia opisu w formie graficznej i jego dokumentacji. Z powodu zbyt krótkiego czasu trwania projektu eksperyment, aż do uzyskania implementacji, kontynuowany był jedynie dla specyfikacji w wersji Estelle.

Specyfikacje w SDL, i w znacznie poważniejszym stopniu w Estelle, poddane zostały weryfikacji metodami symulacyjnymi (a więc poniekąd testowaniu na poziomie specyfikacji). Punktem odniesienia był dokument na temat protokołu [14] oraz równolegle prowadzone prace studyjne nad zaproponowaniem definicji usług XTP (*XTP service*). Na tym etapie struktura specyfikacji bardziej odpowiadała logicznej i funkcjonalnej konstrukcji samego protokołu niż potrzebom implementacji.

Prototypowa implementacja XTP została wygenerowana przy użyciu narzędzi EDT dla środowiska UNIX-owego. Eksperymenty przeprowadzono na stacjach roboczych typu SPARC 5/110.

Podstawowym założeniem projektowym była teza, że otrzymana (prawie) automatycznie implementacja z oryginalnej specyfikacji będzie całkowicie nieefektywna, w szczególności dla potrzeb ruchu multimedialnego. Stąd zrodziło się drugie ważne zadanie projektu: zaproponować i przetestować metodologię analizy wydajnościowej protokołu na poziomie jego specyfikacji, tak by jeszcze przed implementacją przewidzieć jego zachowanie, przy różnej intensywności ruchu w danym węźle i przy różnych obciążeniach pozostałych części sieci. Takie podejście wymagało wymodelowania całego środowiska, w którym implementacja protokołu jest przewidziana (w tym przypadku w Estelle). W szczególności w tym projekcie obejmowało to wymodelownie strumieni ruchu multimedialnego po stronie aplikacji i ruchu w sieci pomiędzy jej węzłami. Jako model ruchu multimedialnego przyjęto kombinację strumieni audio i video, przy czym dla tego drugiego wzorzec stanowił znany format MPEG.

Ogólne warunki metodologiczne, dotyczące powyższych wymagań, zostały sformułowane najpierw maksymalnie niezależnie od rodzaju FDT i jego narzędzi. Bardziej szczegółowy ich opis oraz dokumentacja przeprowadzonych analiz dotyczyła specyfikacji XTP w Estelle.

Ważne w tym eksperymencie było przekonanie się, że przeprowadzone symulacje można stosować (choć nie bez trudności) do rzeczywistych protokołów i uzyskać pewne informacje, dotyczące zachowania się implementacji we wczesnym etapie projektowania. W wyniku przeprowadzonych symulacji (oraz wcześniej znanych cech języka Estelle), pierwotna specyfikacja XTP w kolejnych iteracjach była zmieniana i dostosowywana do wymogów implementacyjnych. Ostatecznie otrzymano prototypową implementację znacznie efektywniejszą (co nie znaczy, że efektywną) od tej, którą można by uzyskać ze specyfikacji w wersji oryginalnej. Co więcej, okazała się ona porównywalna efektywnością z dostępną implementacją XTP otrzymaną poprzez bezpośrednie programowanie w C. Uzyskano także cenne wskazówki, dotyczące stosowanej metodologii i jakości używanych narzędzi.

Podsumowując, wydaje się, że wyniki tego międzynarodowego projektu są zachęcające i wskazują, że praktyczne rezultaty, które można uzyskać, stosując formalne metody inżynierii programowania, są warte zainteresowania.

5. Uwagi końcowe

Niniejszy artykuł próbował usytuować metody formalne zarówno w kontekście historycznym, jak i w odniesieniu do współczesnych uwarunkowań. Celem tych uwag było wskazanie właściwego miejsca tych metod i rozważnej oceny możliwości ich stosowania. Skupienie uwagi na wykorzystaniu metod formalnych w praktyce projektowania protokołów komunikacyjnych pokazuje, że szczególne potrzeby każdej dziedziny i uwzględnienie wiedzy zgromadzonej latami przez specjalistów prowadzi do

metod bardziej przydatnych i efektywnych. Z drugiej strony, obserwacja rozwoju inżynierii oprogramowania w tej szczególnej dziedzinie potwierdza ogólne tendencje. W coraz większym stopniu uwaga projektantów dużych systemów skupia się na początkowych fazach projektowania - one są bowiem decydujące dla jego żywotności i jakości - a więc elementach, które decydują także o efektach ekonomicznych.

Nawiązując na koniec do tytułu tej pracy, można powiedzieć, że mity, jak zwykle, nie znajdują potwierdzenia w rzeczywistości, ale jednocześnie trzeba jeszcze wiele wysiłku, żeby rzeczywistość nie ulegała mitologii.

Literatura

[1]. S. Budkowski, Estelle Development Toolset, Computer Networks and ISDN Systems, Vol. 25, No 1, 1992

[2]. R. Burch, E. M. Clarke, K. L. McMillan, D. L. Dill, L. J. Hwang, Symbolic Model Checking: 10^{20} states and beyond, Information and Computation, 98(2), 1992

[3]. COP 62, High Speed Communication Systems Supporting Multimedia Applications - Selected Issues, Final Project Report COP 62/5, EC INCO Programme, 1997

[4]. J-P. Courtiat, P. Dembiński, G.J. Holzmann, L. Logrippo, H. Rudin, P. Zave, Formal methods after 15 years: Status and trends, Computer Network and ISDN Systems, Vol. 28, No 13, 1996

[5]. M. Flasiński, Wstęp do analitycznych metod projektowania systemów informatycznych, WNT Warszawa, 1997

[6]. O. Haugen (ed.), SDL and MSC, Special Issue, Computer Networks and ISDN Systems, Vol. 29, No 13, 1996

[7]. G. J. Holzmann, Design and Validation of Computer Protocols, Prentice-Hall International, 1991

[8]. ISO 8824, Information processing systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1), Geneva 1987

[9]. ISO 8825, Information processing systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1), Geneva 1987

[10]. ISO 7498, Information processing systems - Open Systems Interconnection - Basic Reference Model, Geneva 1984

[11]. M. Piechówka, S. Szejko, CASE, jaki jest, każdy widzi, Informatyka 4, 1998

[12]. O. Rafiq, A. R. Cavalli (eds.), Protocol Testing, Special Issue, Computer Networks and ISDN Systems, Vol. 28, No 1, 1996

[13]. K. J. Turner (ed.) Using Formal Description Techniques. An Introduction to Estelle, Lotos and SDL, Wiley 1993

[14]. XTP Forum, Xpress Transport Protocol Specification, XTP Revision 4.0, 1995

Janusz Zalewski

Dept. of Electrical & Computer Engineering
University of Central Florida
Orlando, FL 32816, USA
(407)823-6171
jza@ece.engr.ucf.edu

Systemy czasu rzeczywistego – projektowanie i aspekty praktyczne

Streszczenie. *W artykule przedstawiono wybrane elementy charakterystyki systemów czasu rzeczywistego z punktu widzenia projektowania. Omówiono metody specyfikowania i projektowania oraz implementacji tych systemów, uwzględniając elementy języków programowania czasu rzeczywistego, właściwości jądra systemu operacyjnego czasu rzeczywistego i architektury sprzętowe stosowane w takich systemach.*

1. Wprowadzenie

System czasu rzeczywistego jest to system komputerowy o ograniczonym, narzuconym z góry, czasie reakcji. Jego istotą są ograniczenia czasowe, terminy (*ang. deadline*), na wykonanie odpowiednich operacji obliczeniowych. Ten fakt powoduje, że konstruowanie systemów czasu rzeczywistego jest zasadniczo odmienne od konstruowania systemów przeznaczonych do innych zastosowań, np. w zarządzaniu lub handlu. Niezbędne jest więc opracowanie nowych metod wytwarzania systemów, uwzględniających uwarunkowania czasowe.

Ostatnia znana mi książka w języku polskim na ten temat została opublikowana w 1988 roku [17], a bieżące publikacje w miesięczniku *Informatyka*, znakomite w pomyśle i w treści, aczkolwiek nie zawsze aktualne, dowodzą, że jest duże zapotrzebowanie na wiedzę z tej dziedziny. Z pewnością, warto więc polskim czytelnikom przedstawić uzupełniające informacje o aktualnym stanie technologii systemów czasu rzeczywistego.

Z praktycznego punktu widzenia, do przedstawienia wiedzy o systemach czasu rzeczywistego wygodne jest podejście hierarchiczne. Punktem wyjścia jest stwierdzenie i zrozumienie faktu, że wytwarzanie systemów czasu rzeczywistego odbywa się zgodnie z zasadami tradycyjnego cyklu istnienia systemów komputerowych (*ang. system lifecycle*): od specyfikowania, przez projektowanie, do implementowania itd.

Na każdym z tych etapów potrzebny jest, jednak, zasadniczo odmienny rodzaj technologii. Dlatego celowe wydaje się przedstawienie ich w sposób zstępujący, jak na rys. 1, gdzie oprócz poziomu specyfikowania i projektowania, wyróżniono trzy poziomy implementacyjne: języki programowania czasu rzeczywistego, jądro systemu operacyjnego czasu rzeczywistego i architektury sprzętowe.

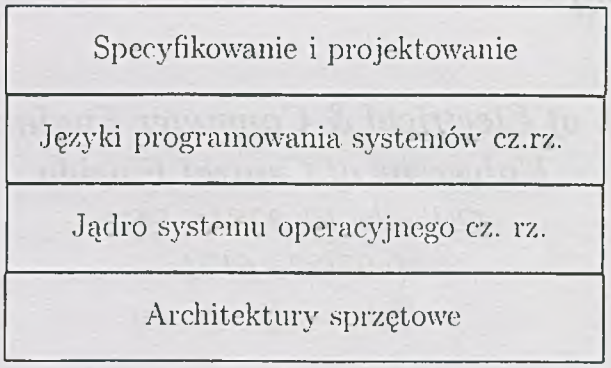


Fig. 1. Hierarchiczne podejście do systemów czasu rzeczywistego.

Taki podział tematyki odpowiada rzeczywistym problemom spotykanym w wytwarzaniu systemów czasu rzeczywistego do poszczególnych zastosowań. W praktyce bowiem, wszystkie rodzaje zastosowań w czasie rzeczywistym, np. systemy komunikacyjne czasu rzeczywistego, bazy danych czasu rzeczywistego, systemy ekspertowe czasu rzeczywistego itp., wymagają bardzo podobnych technologii wytwarzania. Zgodnie z tym podziałem hierarchicznym, artykuł podzielono na części przedstawiające wybrane zagadnienia technologii odpowiadające każdemu z wymienionych poziomów.

2. Specyfikowanie i projektowanie

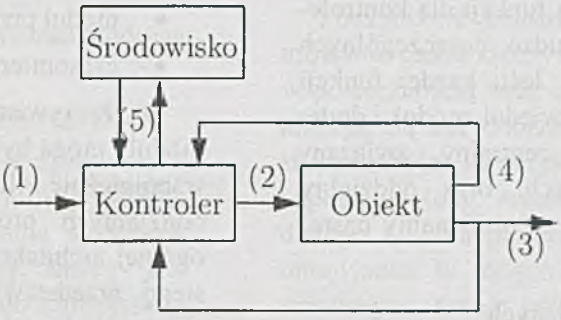
Celem specyfikacji jest, na ogół, wyrażenie właściwości funkcjonalnych konstruowanego systemu. W systemach czasu rzeczywistego, chodzi dodatkowo o wyrażenie właściwości odmiennych od funkcjonalnych, głównie czasowych (zwanych również temporalnymi), a także niezawodnościowych, bezpieczeństwa itp. Aczkolwiek, powstało dotychczas wiele metod specyfikowania systemów czasu rzeczywistego, w szczególności, wiele tzw. metod formalnych [8, 13], niewiele z nich nadaje się do zastosowań praktycznych i w większości pozostają one w stadium eksperymentalnym.

Spośród metod, które uczyniły stosunkowo największy postęp w ostatnich latach warto wymienić następujące:

- logika temporalna [16]
- metody synchroniczne [7]
- systemy hybrydowe [5].

Inne metody formalne, nawet te od dawna rozwijane, jak sieci Petriego lub CCS, dostosowują się do wymagań systemów czasu rzeczywistego bardzo powoli [10, 14].

Zanim jednak metody formalne zaczną odgrywać większą rolę w specyfikowaniu i projektowaniu systemów czasu rzeczywistego, co niewątpliwie nastąpi w przyszłości, inżynierom i projektantom potrzebne są techniki praktyczne. Autor artykułu jest zwolennikiem podejścia inżynierskiego, opartego na pojęciach techniki sterowania, uwzględniającego także metody formalne. Centralnym elementem tego podejścia jest potraktowanie systemu czasu rzeczywistego jako systemu sterowania [9]. Wszystkie znane dotąd autorowi systemy czasu rzeczywistego dadzą się umieścić w tym schemacie, przedstawionym szkicowo na rys. 2.



Rys. 2. Ilustracja podstawowego problemu sterowania: (1) wartość zadana; (2) komendy kontrolera; (3) wielkości sterowane; (4) inne wielkości mierzone; (5) sprzężenie ze środowiskiem

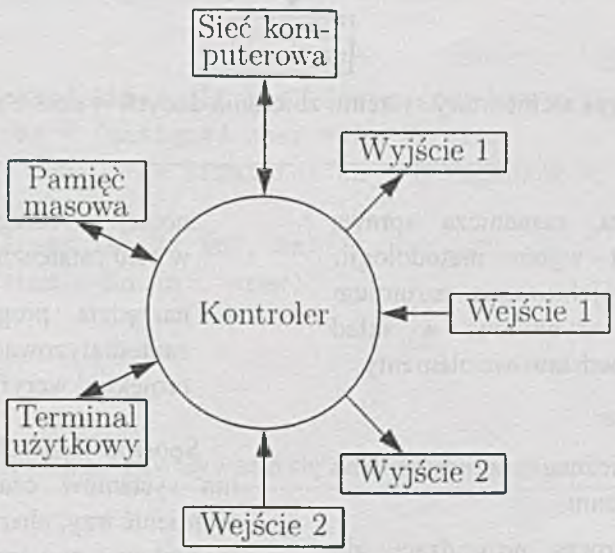
Główne kategorie systemów czasu rzeczywistego pozostają w pełnej zgodności ze schematem na rys. 2, w zależności od połączenia kontrolera z obiektem, tzn.:

- systemy zbierania danych, gdzie połączenie oznaczone przez (2) nie istnieje, ponieważ brakuje na ogół elementów sterowania (np. w systemie monitorowania pacjenta na sali operacyjnej)
- system sterowania programowego, gdzie połączenia oznaczone przez (3) i (4) nie istnieją, gdyż brakuje sprzężenia zwrotnego (np. w sterowniku świateł ulicznych)
- pełny system sterowania, gdzie wszystkie połączenia są konieczne (np. w systemie sterowania lotem samolotu).

Z takiego przedstawienia systemu czasu rzeczywistego łatwo wywnioskować, że kontroler komunikuje się z otoczeniem, za pośrednictwem co najwyżej (zależnie od zastosowania) czterech rodzajów urządzeń zewnętrznych, którymi są:

- wejścia/wyjścia z/do obiektu
- terminal użytkowy
- pamięć masowa
- sieć komputerowa.

Wszystkie te elementy trzeba odpowiednio uwzględnić w projektowaniu, co prowadzi do ogólnego schematu kontekstu kontrolera, przedstawionego na rys. 3.

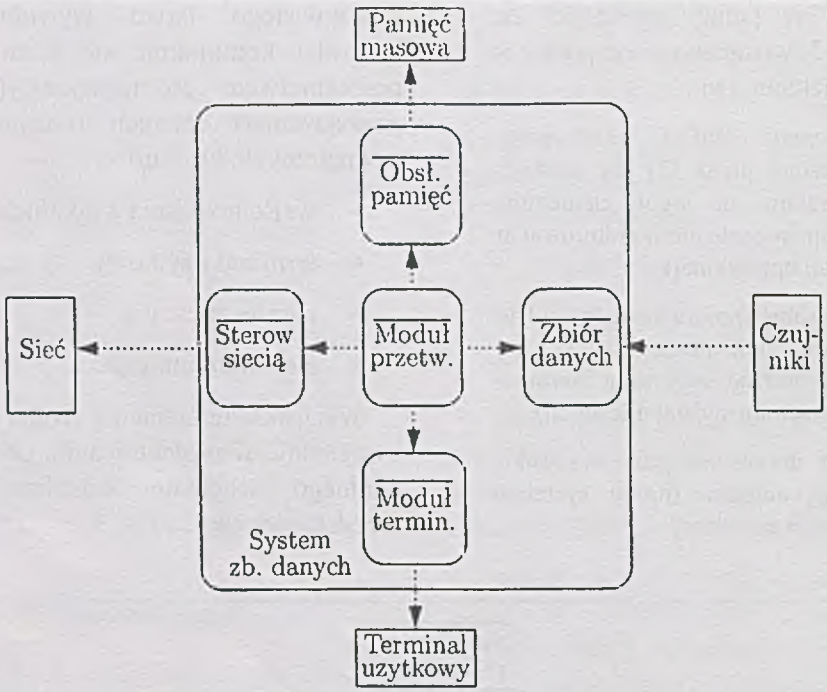


Rys. 3. Ogólny schemat kontekstu kontrolera w systemie czasu rzeczywistego.

Z tak przedstawionego kontekstu, wynika ogólny schemat podziału funkcji dla kontrolera, odpowiadający obsłudze poszczególnych urządzeń zewnętrznych. Jeśli każdej funkcji przyporządkujemy odpowiedni moduł i dodamy niezbędny moduł centralny, związany z przetwarzaniem danych, oraz oddzielny czasomierz (*ang. timer*), to otrzymamy następującą listę:

- moduł zbierania danych
- moduł sterowania
- moduł sprzężenia użytkowego
- sterownik komunikacyjny
- handler pamięci masowej
- moduł przetwarzania danych
- czasomierz.

W rzeczywistych systemach, wymienione moduły mogą być zrealizowane sekwencyjnie, współbieżnie lub nawet równolegle na oddzielnych procesorach. Prowadzi to do ogólnej architektury systemu czasu rzeczywistego, przedstawionej na rys. 4, dla systemu zbierania danych. Taka architektura stanowi bardzo wygodny punkt wyjścia do projektowania systemów czasu rzeczywistego.



Rys. 4. Zarys architektury systemu zbierania danych w czasie rzeczywistym.

Od tego momentu, zasadniczą sprawą w projektowaniu jest wybór metodologii. Przez metodologię projektowania rozumiem zintegrowany zespół czynników, w skład którego wchodzą trzy podstawowe elementy:

- metoda, obejmująca
  - notację do graficznego przedstawienia architektury systemu
  - zdefiniowany proces prowadzący do wytworzenia systemu
- techniki służące do przekształcania graficznej reprezentacji systemu, w ramach notacji i według przyjętych reguł procesu, w celu ostatecznego wytworzenia systemu
- narzędzia programistyczne pozwalające zautomatyzować proces uszczegółowiania projektu i weryfikowania go.

Spośród licznych metodologii projektowania systemów czasu rzeczywistego, warto wymienić trzy, charakteryzujące się tym, że są stosunkowo nowoczesne, tzn. ukierunkowane obiektowo, i kompletne, tzn. obejmujące wszystkie wymienione wyżej składniki: metodę, techniki i narzędzia:

- HRT-HOOD, Hard Real-Time Hierarchical Object-Oriented Design [2]
- ROOM, Real-Time Object-Oriented Modeling [18]
- UML, Universal Modeling Language [3].

HRT-HOOD była jedną z pierwszych metodologii ukierunkowanych obiektowo, przeznaczonych do projektowania systemów czasu rzeczywistego, ale jej zasięg jest ograniczony prawie wyłącznie do Europejskiej Agencji Przestrzeni Kosmicznej (European Space Agency – ESA). Natomiast metoda ROOM zyskała znaczną popularność na rynku amerykańskim, dzięki prostocie i uniwersalności. Większość obecnie rozwijanych metodologii, włączając HRT-HOOD i ROOM zmierza do zgodności ze standardową wersją UML. W niniejszym artykule nie starcza miejsca na, choćby krótkie, omówienie metod i technik, ale warto wymienić niektóre narzędzia programistyczne dla tych metodologii [23]: HRT-HoodNICE, ObjecTime i Rhapsody.

3. Zagadnienia implementacji

Implementowanie systemów czasu rzeczywistego obejmuje trzy zasadnicze zagadnienia: rolę języka programowania, jądro systemu operacyjnego i architekturę sprzętu. Poniżej omówiono krótko tę problematykę.

3.1. Język programowania

Podstawowe konstrukcje języków programowania czasu rzeczywistego są związane ze współbieżnością oraz synchronizacją i komunikacją. Są one dobrze zrozumiane i szeroko stosowane, jak na przykład zadania (*ang. task*) i spotkania (*ang. rendezvous*) w języku Ada, dlatego nie ma potrzeby ich szczegółowego omawiania. W innych językach powszechnie używanych, takich jak C, C++ lub Java, nie ma konstrukcji współbieżnych, dlatego korzysta się z bibliotek lub klas implementujących wielowątkowość (*ang. multithreading*) na poziomie systemu operacyjnego.

Spośród zagadnień szczegółowych, mających kluczowe znaczenie właśnie na poziomie języka programowania, warto wymienić następujące:

(1) dostęp do rejestrów bezpośrednio z języka wysokiego poziomu

(2) obsługa przerwań w języku wysokiego poziomu

(3) mieszanie kodu w różnych językach programowania.

Krótkie przykłady na (1) i (3), dla języka C oraz C i asemblera procesora Sparc, podano, odpowiednio, na rys. 5 i 6.

```
main()
{   // Activate commented lines, if real device reg can't be accessed
    unsigned char * reg = (unsigned char *)0xF7FF0000;
//  unsigned char x = 255;   /* Simulates an I/O register */
//  unsigned char * reg = &x;
//  printf("Legal address is: %x \n", &x);
    printf("Reg contains = %d \n", *reg);
}
```

Rys. 5. Przykład odwoływania się do rejestru w języku C.

```
! Sparc assembly mult function "int mult(int, int)" to be called from C.
! Sample use:  printf("Value returned: %d\n", mult(a, b));
.text
        .align 4
        .global _mult
_mult:
        save %sp,-104,%sp       ! stack frame
        st %i0,[%fp+68]         ! get first argument from call
        st %i1,[%fp+72]         ! get second argument from call
        ld [%fp+68],%o0         ! load first arg for mult
        ld [%fp+72],%o1         ! load second arg for mult
        call .umul,0
        nop
        mov %o0,%i0             ! return value in %i0
        ret
        restore
```

Rys. 6. Przykład podprogramu wywoływanego z języka C, dla procesora Sparc.

Programując systemy czasy rzeczywistego, należy zwrócić szczególną uwagę na użycie konstrukcji, które mogą mieć znaczący wpływ na zależności czasowe i powodować ewentualne negatywne konsekwencje pod tym względem. Do takich konstrukcji należą w szczególności wywołania rekursywne i pętle. O ile rekursji nie zaleca się używać w ogóle w tego rodzaju systemach, o tyle pętle nieskończone stanowią podstawowy element ich programowania, dlatego stosowanie pętli powinno być bardzo ostrożne i świadome. Ogromnie pozytywną rolę w tego rodzaju sytuacjach odgrywa użycie wyjątków (*ang. exception*).

Warto też wyraźnie podkreślić, że wbrew powszechnej opinii, języki zorientowane obiektowo nie są zbyt odpowiednie dla systemów czasu rzeczywistego, ze względu na obecność konstrukcji wykazujących niedeterministyczne zachowanie, takich jak funkcje wirtualne, konstruktory i destruktory. Projektant systemu działającego w czasie rzeczywistym musi gwarantować przewidywalność jego zachowa-nia podczas akcji, a tego nie da się dokładnie przewidzieć przy użyciu wymienionych kon-strukcji. Dlatego nie ma większego sensu programowanie systemów czasu rzeczywiste-go w językach takich jak Smalltalk. Ada pozostaje nadal jedynym prawdziwym języ-kiem do programowania tych systemów.

3.2. Jądro czasu rzeczywistego

Tradycyjne systemy operacyjne nie nadawały się do pracy w czasie rzeczywistym z wielu powodów, z których można wymienić trzy zasadnicze:

- algorytm szeregowania zadań oparty na równoprawności (*ang. fairness*), co jest w konflikcie z zasadą priorytetowej obsługi zadań, podstawowym wymaganiem czasu rzeczywistego
- obsługa przerwań uniemożliwiająca wywłaszczanie (*ang. preemption*) jądra operującego w systemowej przestrzeni adresowej
- organizacja systemu plików oparta na przydzielaniu obszaru przy żądaniu zapisu, zamiast przy tworzeniu pliku, uniemożliwiająca utrzymywanie da-nych w sposób zwarty (*ang. contiguous*).

W związku z tym, pod naciskiem najbardziej wymagających zastosowań, np. w fizyce wysokich energii i w badaniach kosmicznych, na początku lat osiemdziesią-tych powstał cały szereg małych systemów operacyjnych, dostosowanych do pracy

w czasie rzeczywistym. Ich producenci i użytkownicy doszli wkrótce do wniosku, ze należy ujednolicić podstawowe konstrukcje i sposób operowania nimi przez wywołania systemowe [19]. Narzucenie dodatkowego wymagania na zgodność z powłoką (*ang. shell*) systemu operacyjnego Unix, doprowadziło do powstania normy Posix.4 [4] i jej pochodnych. Zgodność z normą Posix.4 (przemianowaną później na Posix.1b) jest najważniejszą charakterystyką współczesnych systemów operacyjnych czasu rzeczywistego, a szczególnie ich wewnętrznej części zwanej jądrem (*ang. kernel*).

Normalizacja ma na celu uzyskanie przenośności kodu źródłowego i dotyczy w zasadzie wszystkich typowych elementów jądra systemu operacyjnego, tj. zarządzania procesami, zarządzania pamięcią i urządzeniami wejścia/wyjścia. Najważniejsze są zalecenia związane z zarządzaniem procesami, które można zgrupować następująco:

- szeregowanie procesów
- komunikacja międzyprocesowa
- obsługa sygnałów czasu rzeczywistego
- czasomierze.

W ramach jednego artykułu, nie można poświęcić wiele miejsca na szczegółowe omówienie tej problematyki, więc pozostaje mi odesłać czytelnika do odpowiedniej literatury [4]. Warto tu jednak wspomnieć o zaleceniach dotyczących szeregowania procesów. Posix.4 definiuje trzy reguły szeregowania, określone przez następujące stałe systemowe:

- SCHED_FIFO, proces wykonuje się do czasu aż dobrowolnie zrzeknie się użycia procesora (chyba, że zostanie wywłaszczony przez proces o wyższym priorytecie)
- SCHED_RR, proces wykonuje się tylko przez określony okres zwany kwantem, po czym oddaje procesor w użytkowanie następnemu procesowi na kolejny kwant czasu itd.
- SCHED_OTHER, reguła zdefiniowana implementacyjnie.

Regułę SCHED_OTHER wprowadzono po części dlatego, aby umożliwić szeregowanie monotoniczne według okresów (*ang. rate monotonic scheduling*) lub według terminów (*ang. deadline monotonic scheduling*). Szeregowanie monotoniczne według okresów dotyczy zadań okresowych, wzajemnie niezależnych, o terminie krótszym niż okres, i polega na priorytetyzacji zadań według długości okresów – *im krótszy okres zadania, tym większy jego priorytet.* W praktyce, podczas wykonywania programu, reguła ta sprowadza się do przydzielania procesora zadaniu, które ma w danej chwili najwyższy priorytet, określony statycznie na podstawie analizy zadań programu.

Metoda ta została opracowana już w latach siedemdziesiątych, ale weszła do powszechnego użytku dopiero stosunkowo niedawno. Jej praktyczna wartość polega na możliwości statycznego sprawdzenia szeregowalności zadań (*ang. schedulability*), za pomocą niezbyt złożonej procedury, biorącej pod uwagę tylko czas wykonania i okres poszczególnych zadań. Warunek wystarczający szeregowalności ma postać następującą:

$$\sum_{i=1}^{N} C_i/T_i \leqslant N(2^{1/N} - 1)$$

gdzie N jest liczbą zadań, a C_i *i* T_i *(i = 1,...,N)* odpowiednio, długością czasu wykonania i okresem zadania *i*. Warunek konieczny i wystarczający szeregowalności jest nieco bardziej skomplikowany. Elementarne omówienie obu warunków, wraz ze szczegółowymi przykładami, można znaleźć w [21]. Bardziej szczegółowa analiza szeregowalności wiąże się z koniecznością zapewnienia odpowiedniej liczby poziomów priorytetów i z przeciwdziałaniem tzw. inwersji priorytetów – nie ma tu jednak miejsca na jej dalszą dyskusję.

Spośród systemów operacyjnych czasu rzeczywistego, z którymi zetknął się autor, spełniającymi zarówno zalecenia normalizacyjne Posix.4, jak również umożliwiającymi – w znacznym stopniu – spełnienie wymagań szeregowania monotonicznego według okresów lub terminów, do najpopularniejszych należy grupa czterech X: LynxOS, QNX, VRTX i VxWorks [24]. Wybór najbardziej odpowiedniego systemu operacyjnego zależy w dużym stopniu od zastosowania, ale

w każdym wypadku warto znać wartości i definicje podstawowych parametrów, do których należą:

- zwłoka obsługi przerwania (*ang. interrupt latency*)
- zwłoka rozprowadzenia przerwania (*ang. dispatch latency*)
- czas przełączania kontekstu
- czas utworzenia/zniszczenia zadania
- czas zamknięcia i otwarcia semafora (*ang. lock/unlock*)
- czasy zapełnienia i opróżnienia kolejki wiadomości (*ang. message queue fill/drain*).

Interesujące porównanie podstawowych charakterystyk podano w pracy [15] i, opartej na tej samej metodzie pomiaru, publikacji w Real-Time Magazine 3/97 [26].

Bardzo ciekawe jest umieszczenie w tym kontekście systemu operacyjnego Windows NT. Dla profesjonalisty jest widoczne gołym okiem, że NT nie jest systemem operacyjnym odpowiednim do zastosowań w czasie rzeczywistym, co chcieliby widzieć użytkownicy. Dzieje się tak z kilku powodów, wyliczonych w zwarty sposób, w numerze 3/97 Real-Time Magazine:

- NT nie zapewnia wystarczającej liczby poziomów priorytetów dla oddzielnych wątków
- NT nie ma w ogóle mechanizmu przeciwdziałającego inwersji priorytetów
- obsługa przerwań nie spełnia silnych ograniczeń czasowych i w niesprzyjających okolicznościach może powodować milisekundowe opóźnienia (dzieje się tak na skutek mechanizmu działania odłożonych wywołań procedur (*ang. deferred procedure calls*)).

Rozwiązaniem aktualnie wchodzącym na rynek jest współistnienie NT i jądra czasu rzeczywistego, na tym samym procesorze. Można to zrobić, w zasadzie, na dwa sposoby:

- wbudowania jądra czasu rzeczywistego w NT i przechwytywania przerwań, co wymaga modyfikacji warstwy abstrakcji sprzętu w NT (*ang. hardware abstraction layer, HAL*)
- wykonywania kodu NT jako jednego z zadań jądra czasu rzeczywistego.

Trzy systemy operacyjne tego rodzaju, to RTX, INTime i Hyperkernel [25].

Ostatnim, niezmiernie ważnym tematem dotyczącym jądra systemu operacyjnego czasu rzeczywistego jest pisanie handlerów urządzeń wejścia-wyjścia. Elementarne omówienie tej problematyki zostało ostatnio opublikowane w [12].

3.3. Architektury sprzętowe

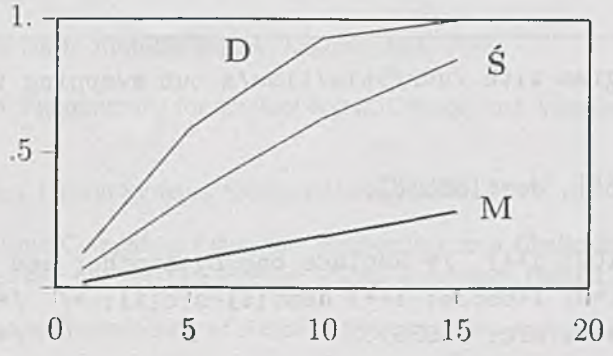
Kluczowym elementem w projektowaniu systemów czasu rzeczywistego jest dokładna znajomość sprzętu. W niniejszym artykule, jest miejsce tylko na zasygnalizowanie wybranych aspektów tej problematyki. We współczesnych zastosowaniach, dwa zagadnienia odgrywają znaczącą rolę, pod tym względem: systemy wieloprocesorowe i rozproszone oraz pamięci podręczne (*ang. cache*).

W systemach wieloprocesorowych i rozproszonych, najważniejszym elementem jest właściwy dobór i dokładna znajomość magistrali lub innego rodzaju połączeń między procesorami systemu [20, 22]. Do zoptymalizowania charakterystyki sprzętu, przed ostatecznym podjęciem decyzji projektowych, stosuje się dwa komplementarne podejścia:

- symulacja rozwiązań istniejących
- wprowadzanie nowych rozwiązań.

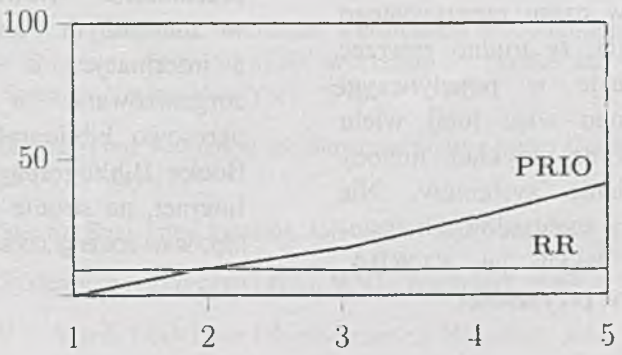
Wśród rozwiązań stosowanych dotychczas, zdecydowanie dominuje użycie magistrali VMEbus. Protokół tej magistrali nie jednak idealnie dostosowany do wymagań systemów czasu rzeczywistego. Symulacja zachowania magistrali VMEbus umożliwia wstępną ocenę jej dopasowania do rodzaju zastosowania pracującego w czasie rzeczywistym. Przykładowo, z wykresów przedstawionych na rys. 7. i 8, wynika że VMEbus działa sprawnie przy liczbie procesorów nie przekraczającej pięć, a arbitraż na zasadzie priorytetu opłaca się stosować w zasadzie tylko wtedy, gdy

należy bezwzględnie zapewnić spełnienie ograniczeń czasowych przez pierwszy procesor [11]. Pozostałe procesory wykazują dłuższy czas dostępu do magistrali niż przy arbitrażu cyklicznym.



Rys. 7. Wykorzystanie magistrali w stosunku do liczby procesorów, dla arbitrażu priorytetowego, przy różnych obciążeniach (duże D, średnie Ś i małe M).

W bardziej wymagających zastosowaniach, magistrale starego typu, takie jak VMEbus, muszą być zastąpione przez nowocześniejsze rozwiązania. Jednym z takich rozwiązań, o parametrach przewyższających tradycyjne magistrale, jest magistrala szeregowa IEEE Std 1394 (FireWire) [1]. Jej przydatność do systemów czasu rzeczywistego wiąże się z wprowadzeniem tzw. transmisji izochronicznej. Mówiąc krótko, polega to na przydzieleniu z góry 80% pasma na transmisje dla urządzeń o wysokim priorytecie. Jeśli żadne z tych urządzeń nie zażąda obsługi w danym cyklu, to pasmo jest wykorzystywane na transmisję asynchroniczną.



Rys. 8. Porównanie zwłoki dostępu do magistrali (ms) względem liczby procesorów dla priorytetowego (PRIO) i cyklicznego (Round-Robin – RR) arbitrażu, przy średnim obciążeniu.

Pamięć podręczna może być znacznym utrudnieniem w zaprojektowaniu systemu czasu rzeczywistego, gdzie zasadniczą rolę odgrywa przewidywalność zachowania pod względem czasowym. Znaczna przypadkowość w dostępie do tego typu pamięci utrudnia analizę zachowania systemu. Na rysunku 9, podano przykład krótkiego programu

ilustrującego trudności związane z określeniem charakterystyk czasowych użycia takiej pamięci. Różnica czasów wykonania pętli (1) i (2) wynosi ponad jeden rząd wielkości na korzyść wersji (2). Można to przewidzieć tylko przy znajomości architektury procesora Sparc i sposobu optymalizacji funkcji `memcpy()`.

```
/* Time this program with /usr/5bin/time/a.out swapping (1) and (2)*/
main()
{
    char src[65536], dest[65536];
    int i, j;
    for (j=0; j<100; j++)  /* Replace one by another and re-run */
        /* for (i=0; i<65536; i++) dest[i]=src[i]; */  /* (1) */
        memcpy(dest, src, 65536);                      /* (2) */
}
```

Rys. 9. Przykład nieprzewidywalności zachowania pamięci podręcznej procesora Sparc.

4. Zakończenie

Jak wskazują wieloletnie tendencje, prędzej czy później, wszystkie systemy komputerowe zostaną zaimplementowane jak systemy czasu rzeczywistego, ponieważ użytkownicy stopniowo zdają sobie sprawę z ich przewagi. Dlatego warto podkreślić znacznie kształcenia profesjonalnego w tej dziedzinie [6].

Tematyka systemów czasu rzeczywistego jest jednak tak rozległa, że trudno zawrzeć wyczerpujące informacje w pojedynczym artykule. Nie poruszono więc tutaj wielu ważnych tematów, jak na przykład, metody i narzędzia uruchamiania systemów. Nie zawarto też informacji o standardowych systemach rozproszonych, takich jak CORBA, które będą dominować w przyszłości. Spośród poruszonej tematyki, wydaje się, ze najlepiej rozwinięte są języki programowania i systemy operacyjne. Nie dopracowano się, natomiast, jeszcze dobrych metod specyfikowania i projektowania systemów czasu rzeczywistego.

Więcej informacji na wymienione tematy można znaleźć w rosnącej literaturze przedmiotu. Główne czasopisma podano w załączonym spisie literatury [26], a informacje o źródłach książkowych zorganizowane w postaci uaktualnianej okresowo bibliografii, „Real-Time Systems: Book Bibliography", są dostępne przez Internet, na stronie sieciowej autora artykułu http://www-ece.engr.ucf.edu/~jza/biblio/rt.biblio.

Literatura

[1] Anderson D., FireWire System Architecture: IEEE 1394, Addison-Wesley, Reading, MA, 1998

[2] Burns A., A. Wellings, HRT-HOOD: A Structured Design Method for Hard Real-Time Systems, Elsevier, Amsterdam, 1995

[3] Douglass B.P., Real-Time UML, Addison-Wesley, Reading, MA, 1998

[4] Gallmeister B.O., Posix.4: Programming for the Real World, O'Reilly and Associates, Sebastopol, CA, 1995

[5] Grossman R.L. et al. (Eds.), Hybrid Systems, Springer-Verlag, Berlin, 1993

[6] Halang W. et al., Real-Time Computing Education: Responding to a Challenge of the Next Century, Real-Time Programming 1997, pp. 121-125, Elsevier Science Publishers, Oxford, 1998

[7] Halbwachs N., Synchronous Programming of Reactive Systems, Kluwer Academic Publishers, Boston, MA, 1993

[8] Hilburn T., J. Zalewski, Real-Time Safety-Critical Systems: An Overview, Proc. 2nd IFAC Workshop on Safety and Reliability in Emerging Control Technologies, pp. 127-138, Elsevier Science Publishers, Oxford, 1996

[9] van Katwijk J., J. Zalewski, Merging Formal Specifications and Engineering Practice in Real-Time Systems Design, Integrated Design and Process Technology, Vol. 6, pp. 57-64, July 1998

[10] Kornecki A., B. Nasah, J. Zalewski, TCAS Safety Analysis Using Timed Environment-Relationship Petri Nets, Proc. Int'l IEEE Symp. on Software Reliability Engineering, ISSRE'98, Paderborn, Germany, 4-7 November 1998

[11] Kornecki A., J. Zalewski, Simulation of Multiprocessor Bus Systems for Real-Time Applications, Proc. SCS Conf. on Simulation Methods and Applications, CSMA'98, Orlando, FL, 1-3 November 1998

[12] Kornecki A. et al., Teaching Device Drivers Technology in Real-Time Systems Curriculum, Proc. 3rd IEEE Workshop on Real-Time Systems Education, Poznań, Poland, 21 November 1998

[13] Langmaack H., W.P. de Roever, J. Vytopil (Eds.), Formal Techniques in Real-Time and Fault-Tolerant Systems, Springer-Verlag, Berlin, 1994

[14] Lichtenecker J., K. Gotthardt, J. Zalewski, Automated Verification of Communication Protocols Using CCS and Binary Decision Diagrams, Proc. 5th IEEE Workshop on Parallel and Distributed Real-Time Systems, pp. 1057-1066, Springer-Verlag, New York, 1998

[15] Low K. et al., Overview of Real-Time Kernels at the Superconducting Super Collider Laboratory, Report SSCL-397, SSC Lab, Dallas, TX, May 1991

[16] Ostroff J.S., Temporal Logic for Real-Time Systems, John Wiley and Sons, New York, 1989

[17] Plaza A.R., E.J. Wróbel, Systemy czasu rzeczywistego, WNT, Warszawa, 1988

[18] Selic B., G. Gullekson, P.T. Ward, Real-Time Object-Oriented Modeling, John Wiley and Sons, New York, 1994

[19] Zalewski J., An Independent Comparison of Three Real-Time Kernel Standards, Paper IEEE P1003.4-N0289, IEEE Posix.4 Meeting, New Orleans, LA, 7-11 January 1991

[20] Zalewski J., Advanced Multimicroprocessor Bus Architectures, IEEE Computer Society Press, Los Alamitos, CA, 1995

[21] Zalewski J., What Every Engineer Needs to Know about Rate Monotonic Scheduling: A Tutorial, pp. 321-335, [20]

[22] Zalewski J., Interconnect Architectures: Old and New, Proc. ESONE Real-Time Data '95 Conf., Warszawa, 27-29 September 1995, pp. 15-21, Przemysłowy Instytut Elektroniki, Warszawa, 1995

[23] Wybrane narzędzia programistyczne do projektowania systemów czasu rzeczywistego:
HRT-HoodNICE
Intecs Sistemi SpA, Via Consalvo 169, Palazzo 8, I-80126 Napoli, Włochy
tel. +39 (81) 242-5711, http://www.intecs.it
ObjecTime
ObjecTime Limited, 340 March Road, Kanata, Ontario K2K 3E4, Kanada,
tel. +1 (613) 591-3535, http://www.objectime.com
Rhapsody
iLogix Inc., 3 Riverside Drive, Andover, MA 01810, USA,
tel. +1 (508) 682-2100 http://www.ilogix.com

[24] Wybrane systemy operacyjne czasu rzeczywistego:
LynxOS
Lynx Real-Time Systems, 2239 Samaritan Drive, San Jose, CA 95124, USA,
tel. +1 (408) 879-3900, http://www.lynx.com
QNX
QNX Software Systems Ltd., Kanata, Ont. K2M 1W8, Canada,
tel. +1 (613) 591-0931, http://www.qnx.com
VxWorks
Wind River Systems, 1010 Atlantic Avenue, Alameda, CA 94501, USA,
tel. +1 (510)748-4100, http://www.wrs.com
VRTX
Mentor Graphics Microtec Div., 880 Ridder Park Drive, San Jose, CA 95131, USA,
tel. (408) 487-7000 http://www.mentorg.com/microtec

[25] Systemy operacyjne czasu rzeczywistego rozszerzające Windows NT:
Hyperkernel
Imagination Systems Inc., 5752 Princess Anne Rd., Virginia Beach, VA 23462, USA,
tel. +1 (757) 518-5400, http://www.imagination.com
INTime
RadiSys, 545 NE Dawson Creek Dr., Hillsboro, OR 98124, USA
tel. +1 (503) 615-1100, http://www.radisys.com
RTX
VenturCom Inc., 215 First St., Cambridge, MA 02142, USA,
tel. +1 (617)661-1230 http://www.vci.com

[26] Zalecane czasopisma profesjonalne:
Informatyka (miesięcznik; cykl „Systemy czasu rzeczywistego"),
Ratuszowa 11, Skr. poczt. 1004, 00-950 Warszawa
ISSN 0542-9951, http://www.pol.pl/informatyka
Embedded Systems Programming (miesięcznik),
525 Market Street, Suite 500, San Francisco, CA 94105, USA,
ISSN 1040-3272, http://www.embedded.com
Real-Time Engineering (kwartalnik),
25875 Jefferson, St. Clair Shores, MI 48081, USA,
ISSN 1076-4429, http://www.realtime-engineering.com
Real-Time Magazine (kwartalnik),
Rue de la Justice 23, B-1070 Brussels, Belgia,
ISSN 1018-0303, http://www.realtime-magazine.com
RTC Magazine (miesięcznik),
27312 Calle Arroyo, San Juan Capistrano, CA 92675, USA,
ISSN 1092-1524, http://www.rtcgroup.com/rtcmag.html
Real-Time Systems (dwumiesięcznik naukowy),
Kluwer Academic Publishers, 101 Philips Drive, Norwell, MA 02061, USA,
ISSN 0922-6443, http://www.wkap.com/journalhome.htm/0922-6443
VMEbus Systems (dwumiesięcznik),
25875 Jefferson, St. Clair Shores, MI 48081, USA,
ISSN 0884-1357, http://www.vmebus-systems.com

Grzegorz Kazimierski

UMBRELLA PROJECT
Grzegorz.Kazimierski@UMBRELLA.org.pl

Wdrażanie systemów jakości zgodnych z ISO 9000

Streszczenie wykładu:

Zarówno w Polsce jaki i na całym świecie obserwuje się coraz większe zainteresowanie projakościowymi metodami zarządzania przedsiębiorstwami.

Posiadanie systemu zapewnienia jakości, potwierdzonego certyfikatem wydanym przez niezależną jednostkę certyfikującą, odgrywa dla wielu polskich przedsiębiorstw coraz większą rolę. Opracowany i wdrożony system jakości świadczy o przygotowaniu przedsiębiorstwa do spełnienia wymagań klienta.

System zapewnienia jakości zgodny z wymaganiami norm serii ISO 9000 to między innymi:

- *Określenie polityki jakości z mierzalnymi i realizowalnymi celami,*
- *Zaangażowanie kierownictwa przedsiębiorstwa i zapewnienie środków,*
- *Opracowane, udokumentowane i wdrożone procedury realizacji procesów wykonywanych w przedsiębiorstwie,*
- *Uruchomione narzędzia ciągłego doskonalenia systemu.*

W prezentacji zostanie przedstawione nowe podejście do jakości, geneza i współcześnie pojmowana rola norm serii ISO 9000, oferta pomocy w opracowaniu i wdrożeniu systemu zapewnienia jakości.

Henryk Krawczyk

Katedra Architektury Systemów Komputerowych
Wydział Elektroniki, Telekomunikacji i Informatyki
Politechnika Gdańska

Modele i ocena jakości systemów informatycznych

Streszczenie

Przedstawiono kontekst wykorzystania wybranych modeli oceny i poprawy jakości systemów i aplikacji informatycznych: QFD, GQM, FP, COCOMO, CMM. Podkreślono znaczenie metrologii oprogramowania oraz potrzebę śledzenia ryzyka podczas realizacji projektów informatycznych. Zasygnalizowano pewne problemy związane z rokiem 2000.

1. Wprowadzenie

Przez system informatyczny rozumiemy każdy system komputerowy wykonujący zadane aplikacje lub oferujący określone usługi informatyczne. Nowoczesne systemy tego typu integrują funkcje przetwarzania, przesyłania, gromadzenia i wizualizacji informacji i są stosowane w różnych dziedzinach życia. Wysoka jakość jest, oprócz wymagań funkcjonalnych, podstawowym kryterium ich projektowania. Istotne staje się więc zrozumienie czynników wpływających na poprawę jakości aplikacji i usług informatycznych oferowanych przez różnego typu systemy komputerowe. Dużą pomocą są tutaj opracowane standardy jakościowe [3]. Przykłady takich standardów przedstawia tabela 1. Podają one jednak dość ogólne wytyczne, które mogą być dość dowolnie interpretowane i implementowane. Istnieje zatem potrzeba definiowania bardziej konkretnych modeli oceny jakości systemów informatycznych. W pracy przedstawiono kontekst wykorzystania szeroko stosowanych modeli praktycznych takich jak: QFD (Quality Function Deplayment [16]), GQM (Goal Question Metric [13]), FP (Function Points [2]), COCOMO (Constructive Cost Model [4]), czy CMM (Capability Maturity Model [7]). Podkreślono znaczenie metrologii oprogramowania oraz śledzenie ryzyka podczas realizacji projektów informatycznych. Zasygnalizowano pewne problemy wiarygodności funkcjonowania systemów związane z rokiem 2000.

Jakość jest rozumiana jako ogół cech i własności usług systemu informatycznego oferowanych dla zaspokojenia wymaganych potrzeb. Bezpośrednio wynika ona z oceny wykorzystywanych aplikacji w systemie. Z kolei sterowanie jakością czy zapewnienie jakości obejmuje wszystkie aspekty działalności organizacji (rozumianej jako spółka, przedsiębiorstwo czy instytucja), a mianowicie dobór, nadzór i kontrolę nie tylko procesów czysto produkcyjnych, ale również prac projektowych, usługowych i organizacyjnych (zawieranie umów, dobór kadr, współpraca z kontrahentami). Trudno jest w jednej pracy przedstawić te wszystkie aspekty, dlatego świadomie ograniczono się poniżej tylko do niektórych problemów, a mianowicie relacji pomiędzy wymaganiami a atrybutami jakości, zasad formułowania uogólnionych reguł jakościowych oraz znaczenia iteracyjnego projektowania sterowanego jakością.

Tabela 1 Przykłady standardów jakościowych

Nazwa standardu	Przeznaczenie
ISO 900x	x = 0 Zarządzanie i zapewnienie jakości. Wytyczne wyboru i stosowania danego modelu x = 1 Pełny model zapewnienia jakości w produkcji, instalowaniu i serwisie x = 2 Zmodyfikowany model zapewnienia jakości x = 3 Uproszczony model zapewnienia jakości x = 4 Zarządzanie jakością i elementy systemu jakości
ANSI/IEEE 1016 ISO/IEC 12207	Procesy cyklu życia oprogramowania
ISO/IEC 9126, 15504	Pomiary i ocena jakości oprogramowania
ANSI/IEEE 1012, 1028	Problematyka testowania i inspekcji oprogramowania

Tabela 2 Wymagania a atrybuty jakości

Wymagania	Atrybuty jakości
Funkcjonalne	funkcjonalność, użyteczność
Czasowe	wydajność
Niezawodnościowe	wiarygodność
Ograniczenie niefunkcjonalne	elastyczność

2. Wymagania a atrybuty jakości

Ocena jakości systemów informatycznych jest wymagana już na etapie wyboru różnych ofert projektów, czy gotowych systemów do zainstalowania. Sprowadza się ona na ogół do optymalnego dopasowania wymagań użytkownika do odpowiednich parametrów jakości odpowiadających jednemu z oferowanych systemów. Skojarzenie wymagań i atrybutów jakości przedstawia tabela 2, gdzie przedstawiono typowe wymagania funkcjonalne i niefunkcjonalne oraz odpowiadające im podstawowe atrybuty jakości. Jednak bez szczegółowego sprecyzowania zarówno wymagań jak i atrybutów jakościowych ostateczne decyzje mogą być trudne nawet dla wybitnych ekspertów.

Tabela 3 Zgrubna metoda oceny ofert

Etapy analizy ofert	Kryteria oceny
1. Wstępna selekcja	Funkcjonalność, renoma firmy
2. Wstępna eliminacja	Koszt, wynik negocjacji
3. Ostateczne decyzje	wybrane atrybuty jakości

Tabela 4 Dokładniejsza metoda oceny - QFD

		1	2	3	4	5	6	7	8	9
		Atrybuty jakości								
		Funkcjonalność	Użyteczność	Wydajność	Wiarygodność	Elastyczność	Waga ważności	Analizowana Firma	Konkurent 1	Konkurent 2	Plan firmy	Współ. udoskonaleń	Współ. sprzedaży	Wagi ogólne	% rozkład wag
Zadania klienta	Funkcjonalne	9	3	1	1	3	5	4	4	5	5	1	1.5	7.5	32
	Czasowe	3	3	9	3	1	2	3	4	3	4	2	1.2	4.8	20
	Niezawodność	1	9	3	9	1	3	3	5	2	5	1.33	1.0	4.0	16
	Inne (koszt)	3	3	3	9	9	4	3	5	3	5	1.25	1.5	7.5	32
	Ogółem	16	18	16	22	14	86							23.8	100%
	%	18.6	20.9	18.6	25.6	16.3	100%								
	Firma														
	Konkurent 1														
	Konkurent 2														
	Plan														

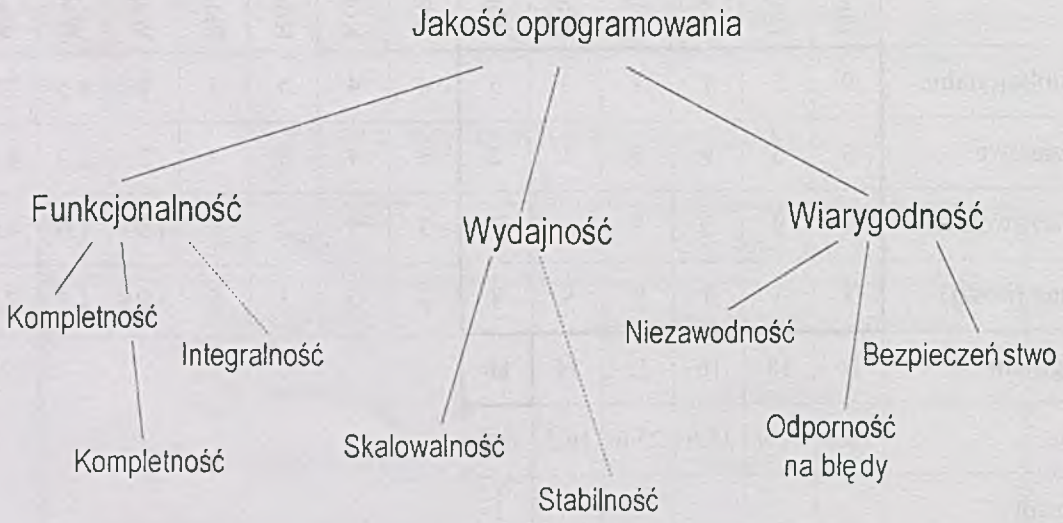
Analiza i Pomiary

Korelacje: silna (9), średnia (3), słaba (1)

Wagi ważności: małoistotna (1) ÷ najważniejsza (5)

Rodzaj sprzedaży: wielka (1.5), średnia (1.2), mała (1)

W praktyce doboru ofert wyróżniamy dwa główne podejścia. Pierwsze polega na zgrubnej ocenie ofert w trzech etapach, w których stopniowo eliminowane są oferty nie spełniające zadanych kryteriów. Tabela 3 podaje przykład tego typu postępowania. Z uwagi na przybliżoną ocenę, dużą rolę mogą odgrywać pewne nieformalne układy. Drugie podejście związane jest już z bardziej wnikliwą oceną ofert. Wykorzystuje się tutaj model QFD, którego prosty przykład znajduje się w tabeli 4. Tego typu podejście stara się określić powiązania wymagań użytkownika z atrybutami jakości za pomocą dwuwymiarowej macierzy (zaznaczono ją linią pogrubioną), określając przy tym trzy różne stopnie korelacji. Tego typu oszacowanie można wykonać dla każdej oferty, uwzględniając lub nie odpowiednie wagi ważności wymagań w skali pięcio punktowej. Bazując na takiej macierzy określić możemy również obiektywny wskaźnik ilościowy zgodności oferty z wymaganiami. Dodatkowo można uwzględnić procentowy rozkład spełnienia wymagań lub procentowy rozkład zapewnienia atrybutów jakości. Umożliwia to znacznie pełniejszą ocenę ofert w porównaniu do metody zgrubnej. Model QFD może być także wykorzystywany do opracowania strategii rozwojowej firmy, poprzez porównanie własnych osiągnięć jakościowych z osiągnięciami konkurentów. Zakładając odpowiedni współczynnik udoskonaleń i współczynnik sprzedaży wytworzonych produktów możemy określić wymaganą poprawę technologii wytwarzania w celu uzyskania przewagi nad konkurentami. W tym przypadku w tabeli 4 zaprezentowano jedynie porównawczą ocenę i plan rozwojowy firmy, pomijając natomiast pełne uwzględnienie poszczególnych atrybutów jakości.



Rys. 1. Przykład drzewa jakości

3. Definiowanie modelu jakości

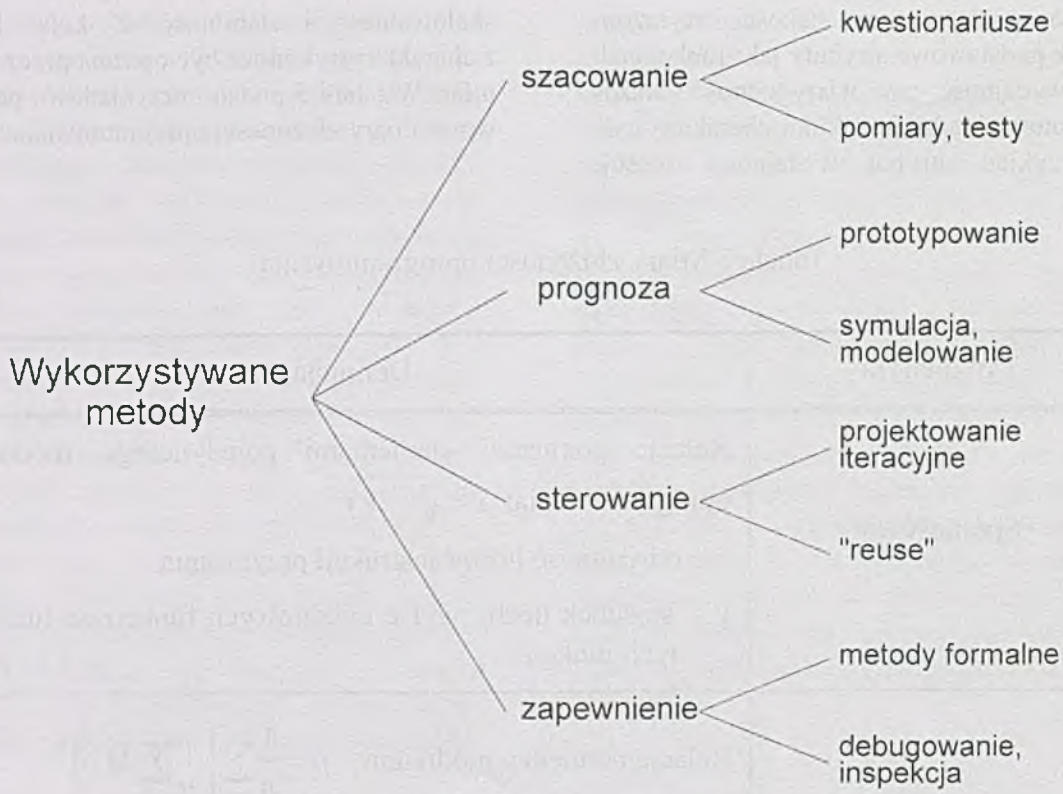
Z rozważań przeprowadzonych w poprzednich rozdziałach wynika, że istotną rolę w procesie analizy jakościowej odgrywa znajomość modelu jakości [19]. Dotychczas nie zdefiniowano uniwersalnego modelu, który byłby akceptowalny dla różnego typu aplikacji i różnych architektur systemów informatycznych. Na ogół model taki definiuje się dla konkretnych potrzeb, przyjmując wielopoziomową strukturę opisu parametrów jakościowych [5]. Przykład takiej struktury przedstawia rys.1. Jest to typowe drzewko jakości (w ogólności diagram relacji czy współzależności), gdzie korzeń stanowi

zintegrowany parametr jakości wyrażony poprzez podstawowe atrybuty jak: funkcjonalność, wydajność czy wiarygodność. Każdy z atrybutów składa się z kilku charakterystyk. Na przykład atrybut wydajności opisuje skalowalność i stabilność. Z kolei każda z charakterystyk może być opisana przez wiele miar. W tabeli 5 podano przykładowo podstawowe miary złożoności oprogramowania.

Tabela 5 Miary złożoności oprogramowania

Typ metryki	Definicja
Spójność (s)	Relacje pomiędzy elementami pojedynczego modułu oprogramowania; $s = \sqrt{x^2 + y^2}$ x - odwrotność liczby instrukcji przypisania y - stosunek liczby wyjść unikatowych funkcji do liczby tych funkcji
Powiązanie (p)	Relacje pomiędzy modułami; $p = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{n}\sum_{j=1}^{n} M_{ij}\right)$ n - liczba modułów M_{ij} - liczba wyjść i-tego modułu współdziclona z wejściami j-tego modułu
Modularność (m)	Określa ziarnistość oprogramowania; $m = \frac{T_p}{T_c}$ T_p - czas wykonania obliczeń zawartych w modułach T_c - czas komunikacji międzymodułowej
Złożoność modułu (z)	Średnia wielkość modułu; $z = \frac{NLOC}{n}$ $NLOC$ - liczba linii kodu oprogramowania

Takie drzewko jakości może być definiowane przez użytkownika przy odpowiednim wsparciu ekspertów. Fakt ten na rys.1 zilustrowano dwoma typami krawędzi. Krawędzie ciągłe są definiowane przez użytkowników, zaś przerywane są dodawane przez ekspertów w celu dokonania pełniejszej oceny. Modyfikacja drzewa jakości jest konieczna w celu zniwelowania różnego poziomu wykształcenia informatycznego potencjalnych użytkowników.



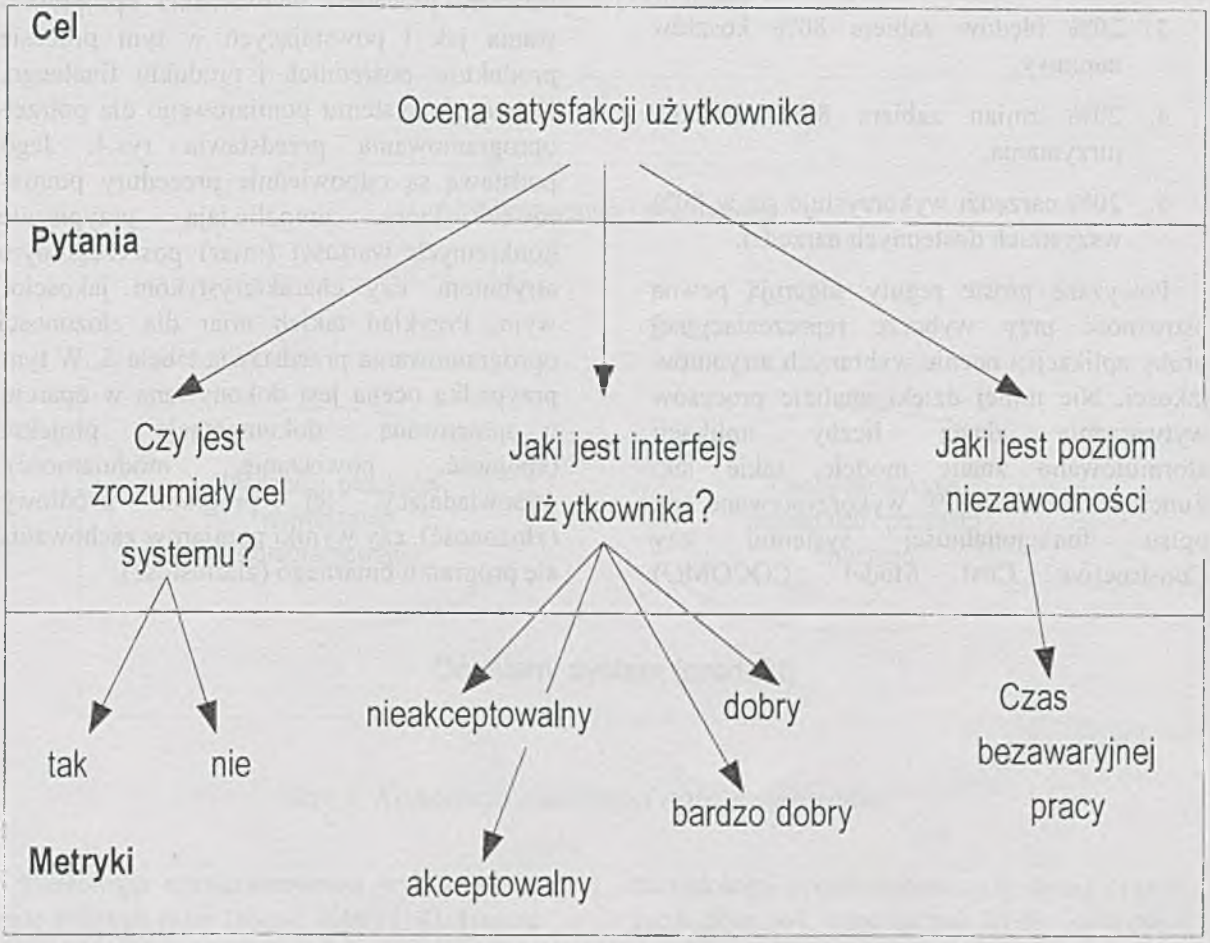
Rys.2. Podstawowe metody oceny i poprawy jakości oprogramowania

Definiowanie drzewka jakości może być warunkiem wyjściowym dokonania oceny [7]. Warunkiem niezbędnym jest badanie odpowiednich procedur wyznaczania miar i charakterystyk jakościowych. Tutaj rola ekspertów jest już raczej nieodzowna. Na rys.2 przedstawiono podstawowe metody umożliwiające pomiar metryk lub poprawę wybranych charakterystyk jakości. Pierwsze z nich dotyczą szacowania jakości produktów informatycznych, co realizuje się między innymi poprzez kompleksowe testowanie lub niezależne opiniowanie. W fazie projektowania bardziej istotne są metody prognozowania, które sprowadzają się do prototypowania bądź symulacji. Sterowanie jakością polega z kolei na stopniowym udoskonalaniu wytwarzanego produktu, co może być dokonywane w iteracyjnym projektowaniu [11] bądź dzięki różnym technikom "reuse" [2]. Zapewnienie jakości polega na otrzymaniu zadanych parametrów wytwarzanego produktu, do czego na ogół niezbędne są metody formalne [9] (np. dowodzenie poprawności czy analiza bezpieczeństwa), lub metody inspekcji kodu bądź procedury eliminowanie istniejących błędów [10,14]. Ponownie, z uwagi na obszerność problematyki, w następnych rozdziałach ograniczymy się jedynie do wybranych zagadnień oceny i zapewnienia jakości.

4. Sondaże i statystyki

Istnieją dwa podstawowe sposoby wyznaczania jakości produktów informatycznych i ich metod wytwarzania. Jeden sposób obejmuje metody eksperymentalne, w których systemy ocenia się na podstawie różnego typu wykonywanych eksperymentów pomiarowych umożliwiających określone wymagania zachowania się systemu oraz identyfikację istniejących powiązań w systemie. Drugi sposób dotyczy metod sprowadzających się do obserwacji systemu i obejmuje ocenę jego zachowania się na podstawie fazy użytkowania. W tym drugim przypadku wnioski są bardziej ogólne, gdyż obserwując biernie, wpływamy w minimalnym stopniu na funkcjonowanie systemu. Typowym podejściem jest przeprowadzenie różnego typu sondaży poprzez wypełnianie odpowiednich kwestionariuszy. Istotne jest by zbierane w takiej formie informacje były reprezentatywne i w miarę obiektywne. Zasady ankietyzowania użytkowników określa metoda GQM (Goal-

Question-Metrics). Prosty przykład jej realizacji pokazany jest na rys.3. Składa się ona z trzech poziomów. Pierwszy definiuje główny cel badania, w rozpatrywanym przypadku oceniana jest satysfakcja użytkownika z pewnego produktu informatycznego. Drugi poziom badań składa się z różnego rodzaju pytań, precyzujących dokładnie przedmiot oceny, w tym analizowane charakterystyki jak: zrozumiałość funkcji systemu, jego interfejs oraz niezawodność. Trzeci poziom dostarcza nam metryk jakościowych, które mogą przyjmować różne wartości: binarne (0,1), wielowartościowe (nieakceptowany, akceptowalny, dobry, bardzo dobry), bądź konkretne wartości określające jeden z parametrów (np. niezawodność wyrażoną poprzez średni czas bezawaryjnej pracy). Zebrane w ten sposób informacje umożliwiają budowę drzewa jakości, wyznaczenie poszczególnych wag charakterystyk jakościowych i mogą być wykorzystane w dalszej analizie, np. QFD.



Rys.3. Przykład implementacji metody GQM dla oceny satysfakcji użytkowania

Analiza jakości może dotyczyć serii wytwarzanych aplikacji, bądź tzw. reprezentatywnego zbioru aplikacji. Zbiór taki zawiera programy różnego typu. Na ogół w jego skład wchodzą programy baz danych, komunikacyjne, interfejsy użytkownika, a także funkcje systemu operacyjnego czy procedury biblioteczne. W oparciu o tak reprezentatywną próbę można dokonać takiej oceny, która z zadanego zbioru parametrów oprogramowania wybierze tylko takie metryki, które w zasadniczym stopniu wpływają na jego atrybuty jakości. Przeprowadzone eksperymenty [18] pokazują na przykład, że spośród następujących parametrów: kategorie programu, częstość zmian specyfikacji, zręczność programisty, złożoność kodu programu (KLOC), organizacje zespołów programistów, technologia programowania, wysiłek programisty, modyfikowalność modułów, tylko cztery pierwsze z nich mają najistotniejszy wpływ na liczbę błędów

residualnych (nie wykrytych w fazie testowania).

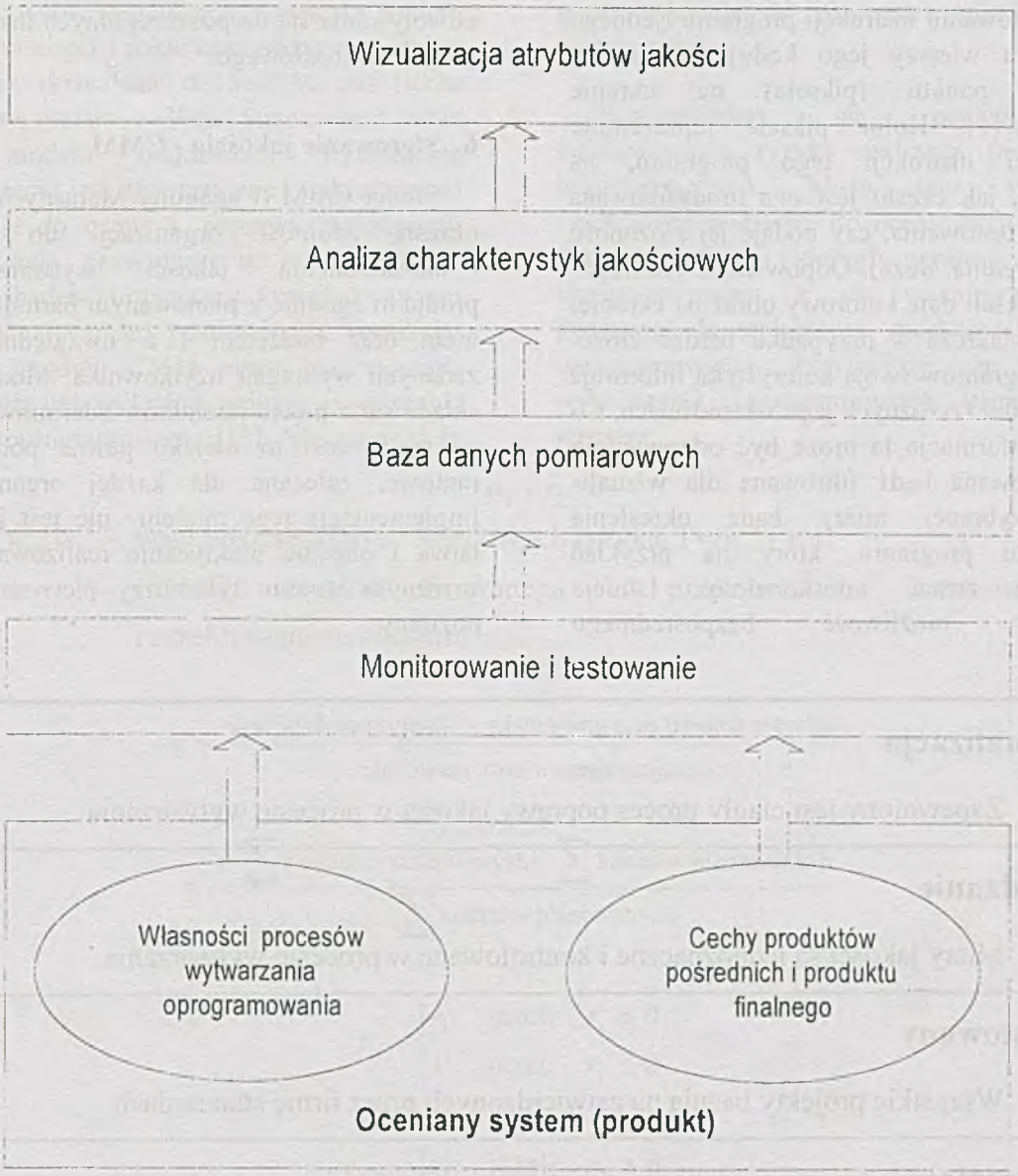
Statystyczna analiza różnego typu pomiarów lub sondaży umożliwia sformułowanie ogólnych reguł, z których najbardziej znaną i często wykorzystywaną jest tzw. rozkład Pareto. Stwierdza on, że w typowych aplikacjach:

1. 20% modułów wykorzystuje 80% resursów,
2. modułów zawiera 80% błędów,
3. 20% błędów zabiera 80% kosztów naprawy,
4. 20% zmian zabiera 80% kosztów utrzymania,
5. 20% narzędzi wykorzystuje się w 80% wszystkich dostępnych narzędzi.

Powyższe proste reguły sugerują pewną ostrożność przy wyborze reprezentacyjnej próby aplikacji i ocenie wybranych atrybutów jakości. Nie mniej dzięki analizie procesów wytwarzania dużej liczby aplikacji sformułowano znane modele, takie jak: Function Points (FP) wykorzystywane do opisu funkcjonalności systemu, czy Constructive Cost Model (COCOMO) umożliwiający szacowanie kosztów oprogramowania na etapie jego projektowania.

5. Metrologia oprogramowania

Przez metrologię oprogramowania rozumie się określone metody i techniki wykorzystania miar w celu otrzymania zamierzonego celu. Na ogół takim celem jest zapewnienie odpowiedniej jakości wytwarzanemu produktowi. Osiągamy go przez ocenę parametrów zarówno procesów wytwarzania oprogramowania jak i powstających w tym procesie produktów pośrednich i produktu finalnego. Koncepcję systemu pomiarowego dla potrzeb oprogramowania przedstawia rys.4. Jego podstawą są odpowiednie procedury pomiarowe, które umożliwiają przypisanie konkretnych wartości (miar) poszczególnym atrybutom, czy charakterystykom jakościowym. Przykład takich miar dla złożoności oprogramowania przedstawia tabela 5. W tym przypadku ocena jest dokonywana w oparciu o opracowaną dokumentację projektu (spójność, powiązanie, modularność), odpowiadający jej program źródłowy (złożoność), czy wyniki pomiarów zachowania się programu binarnego (ziarnistość).



Rys.4. Koncepcja metrologii oprogramowania

Metrologia oprogramowania wykorzystuje wiele różnych miar (ponad 3000) [14]. Istotne więc jest ich odpowiednie ulokowanie w drzewie jakości (patrz rys.1). Tak więc technika, analiza i pomiary systemów prowadzą do konstrukcji drzewa jakości z dołu do góry, bądź z góry do dołu. W każdym z tych przypadków istotne jest przyjęcie uzgodnionych procedur oceny jakości. Poza tym oprócz odpowiednich strategii monitorowania i testowania, istotne jest przechowywanie odpowiednich danych historycznych zebranych dla wielu projektów. Analiza takich danych umożliwia prognozowanie uogólnionych tendencji typowych dla danej metodologii projektowania, czy danej organizacji. Stanowi więc istotne źródło informacji dla ulepszania metodologii wytwarzania, jest alternatywnym rozwiązaniem w stosunku do różnych badań sondażowych. Głównym problemem jest i będzie jednak ogólna dostępność do tego typu informacji.

Wizualizacja parametrów oprogramowania, w tym różnego typu atrybutów, charakterystyk i miar jakości decyduje zasadniczo o przydatności różnego typu narzędzi pomiarowych. Oprócz standardowych rozwiązań jak wykresy czy diagramy wykorzystuje się tu również bardziej oryginalne podejścia. Jedno z nich polega na

reprezentowaniu instrukcji programu (jednego lub kilku wierszy jego kodu) za pomocą jednego punktu (piksela) na ekranie monitora [1]. Kolor piksela reprezentuje własności instrukcji tego programu, na przykład, jak często jest ona modyfikowana podczas testowania, czy podaje jej złożoność (mała, średnia, duża). Odpowiednie uszeregowanie pikseli daje kolorowy obraz na ekranie, który zwłaszcza w przypadku bardzo złożonych programów swoją kolorystyką informuje bezpośrednio o różnych jego własnościach. Co więcej informacja ta może być odpowiednio kondensowana bądź filtrowana dla wizualizacji wybranej miary bądź określenia fragmentu programu, który na przykład powinien zostać udoskonalony. Istnieje wówczas możliwość bezpośredniego odwoływania się do poszczególnych instrukcji programu źródłowego.

6. Sterowanie jakością - CMM

Model CMM (Capability Maturity Model) określa zdolność organizacji do oceny i udoskonalenia jakości wytwarzanego produktu zgodnie z planowanym harmonogramem oraz budżetem i z uwzględnieniem zadanych wymagań użytkownika. Model ten składa się z pięciu poziomów zdefiniowanych na rys.5. Jest to niejako pewne podejście ramowe, zalecane dla każdej organizacji. Implementacja tego modelu nie jest jednak łatwa i obecnie praktycznie realizowane są w różnym stopniu tylko trzy pierwsze jego poziomy.

Optymalizacja Zapewniony jest ciągły proces poprawy jakości w procesie wytwarzania
Zarządzanie Miary jakości są jednoznaczne i kontrolowane w procesie wytwarzania
Zdefiniowany Wszystkie projekty bazują na zatwierdzonych przez firmę standardach
Powtarzalny Dokumentowanie postępowania z możliwością powtórzenia sukcesu dla podobnych aplikacji
Wstępny Procesy wytwarzania przypadkowe i charakterystyczne, indywidualność postępowania

Rys.5. Poziomy dojrzałości procesu oceny i zarządzania jakością oprogramowania

Dyskusję dotyczącą modelu CMM przeprowadzono również w oparciu o ankietyzację osób starających się wprowadzić go w swojej organizacji. Zauważono, że czas przejścia z poziomu 1 do poziomu 2 lub z poziomu 2 do poziomu 3 wahał się od 18 do 30 miesięcy, średnia wynosiła około 2 lat. Natomiast średni koszt wymagany do realizacji zmiany poziomów dojrzałości, zależny od wielu parametrów, wynosił około 1.375 $ w przeliczeniu na jednego inżyniera oprogramowania. Co więcej dzięki zastosowaniu zaleceń CMM,

produktywność organizacji wzrastała średnio o 35% w ciągu 1 roku, czas oddania produktu na rynek skracał się o 15÷23 %, zaś liczba defektów malała o 39%. Sugerowaną wadą tego modelu (najczęściej wymienianą w ankietach) jest zbyt sztywne (niekreatywne) podejście do oceny i poprawy metodologii wytwarzania, prowadzące do wzrostu biurokracji w badanej organizacji. Przeciwieństwem do schematycznego postępowania wynikającego z modelu CMM może być strategia kreowania indywidualnej polityki zwiększania jakości oprogramowania [13]. Nie oznacza to wcale odejścia od pewnych stałych reguł postępowania.

Skoncentrujmy się poniżej na monitorowaniu ryzyka realizacji projektów informatycznych. Może ono stanowić obiektywną podstawę do oceny przydatności różnych metod i strategii poprawy jakości oprogramowania, a co ważniejsze być podstawą do dokonania modyfikacji tego oprogramowania. Przyjmijmy, że ryzyko wytwarzania oprogramowania wyraża się wzorem

$$r = r_1 \cdot w_1 + r_2 \cdot w_2$$

gdzie: w_1, w_2 - wagi poszczególnych składowych ryzyka, $1 = w_1 + w_2$

zaś: r_1, r_2 - wartości ryzyka związanego z przekroczeniem harmonogramu i przekroczeniem budżetu.

Niech

$$r_1' = \frac{\text{czas trwania projektu - planowany czas trwania projektu}}{\text{planowany czas trwania projektu}}$$

$$r_2' = \frac{\sum \text{kosztów rzeczywistych} - \sum \text{kosztów planowanych}}{\sum \text{kosztów planowanych}}$$

Zatem:

$$r_1 = \begin{cases} r_1' & \text{jeżeli } r_1' > 0 \\ 0 & \text{jeżeli } r_1' \le 0 \end{cases}$$

$$r_2 = \begin{cases} r_2' & \text{jeżeli } r_2' > 0 \\ 0 & \text{jeżeli } r_2' \le 0 \end{cases}$$

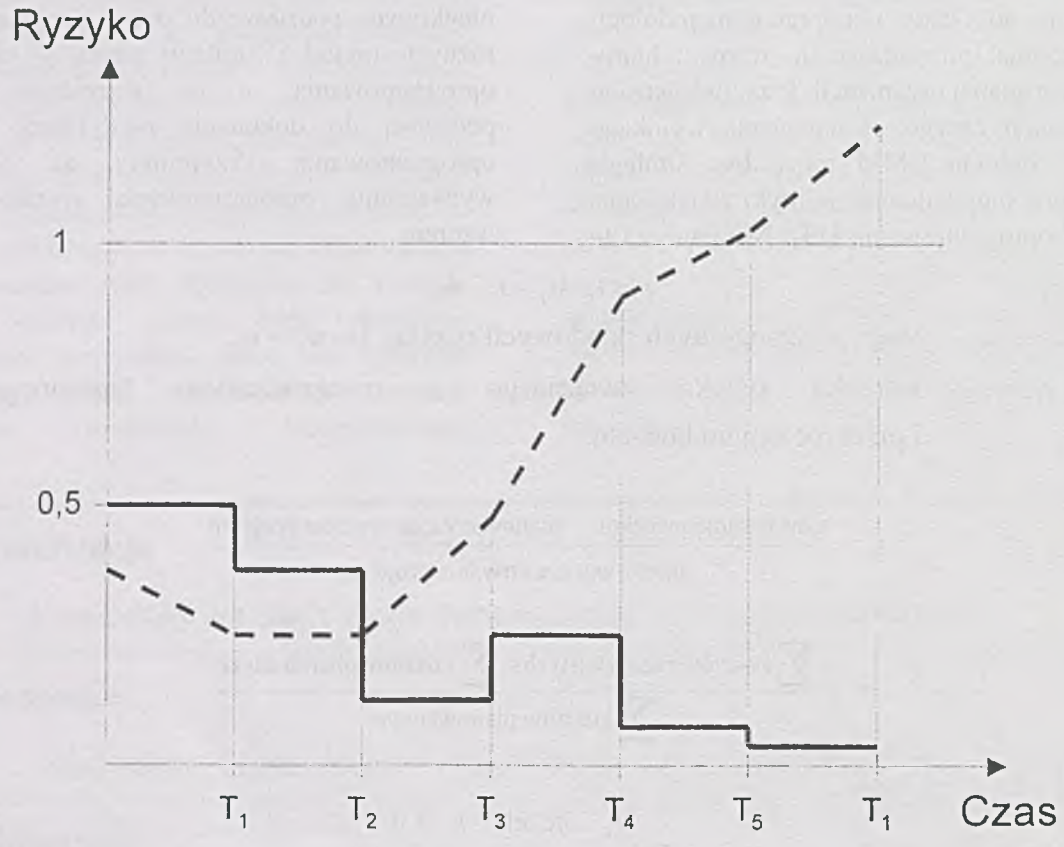
Rys.6 przedstawia wykres zmienności ryzyka r_1 i r_2 w czasie. Jak wynika z rysunku harmonogram analizowanego projektu nie odbiegał zbyt dużo od założonego, zaś nakład kosztów znacznie przekroczył oczekiwania. Ukazanie takich tendencji zmian umożliwia podjęcie decyzji na bieżąco i dokonanie korekty zadań w realizowanym projekcie. Stanowi to punkt wyjściowy realizacji czwartego poziomu modelu CMM.

7. Problemy roku 2000 - Y2K

Jakość systemów informatycznych, a w szczególności wiarygodność ich funkcjonowania może zależeć od wykorzystywanego formatu daty. Szacuje się, że wiele milionów aplikacji wymaga wprowadzenia poprawek w związku z rokiem 2000. Tego typu zagadnienia są ostatnio szeroko dyskutowane jako problem Y2K.

Generalnie w systemach informatycznych czas kalendarzowy jest zapamiętywany w pewnym formacie (np. w Europie rok, miesiąc, dzień) lub określany przez okres czasu (sekundy, tygodnie), który upływa od zadanego momentu. Sposób przedstawienia daty może być różny w różnych aplikacjach. Co więcej w wielu z nich format roku przedstawiany jest za pomocą tylko dwóch cyfr, zaś licznik czasu składa się nie więcej niż z czterech bajtów. W konsekwencji,

w przypadku pierwszym, rok 2000 będzie przedstawiany jako 00, zaś w przypadku drugim licznik czasu wyzeruje się po 1024 tygodniach, tj. co około 20 lat. Może to mieć różne konsekwencje, na ogół prowadzić będzie do błędnego (nieokreślonego) zachowania się aplikacji, zwłaszcza takich których działanie jest zależne od wartości reprezentującej datę.



Rys.6. Monitorowanie ryzyka przekroczenia harmonogramu (linia przerywana) i kosztów (linia ciągła), T_i - okresy kontroli

Istotny jest również sposób wyznaczania roku przestępnego. Odpowiedni algorytm określa, że jest to rok, którego wartość liczbowa dzieli się przez 4, ale nie jest podzielna przez 100, chyba że jest podzielna przez 400. Ostatnio rok 1600 był takim właśnie rokiem, dla którego uwzględnić należało te wszystkie warunki, niedługo będzie nim rok 2000. Dlatego też w algorytmach określania roku przestępnego często pomijano sprawdzenie podzielności przez 400 i w konsekwencji rok 2000 może nie zostać zaliczony do roku przestępnego.

W związku z powyższym test sprawdzający odporność aplikacji na problem Y2K polega na wykonaniu następujących testów:

- test linii granicznych, tj. ocena zachowania się aplikacji na przełomie roku, tj. 31.12.1998 na 1.01.1999, 31.12.1999 na 1.01.2000 oraz 31.12.2000 na 1.01.2001,
- test roku przejściowego, w tym zmiany dat 28.02.2000 na 29.02.2000 oraz 29.02.2000 na 1.03.2000,
- test struktur danych (baza danych, system plików, pola stałe i robocze, raporty), które nie różnią się od wykorzystywanych formatów dat. Na przykład jeśli oznaczenie końca pliku oznaczono jako ciąg 99999 to może zostać on zinterpretowany jako data 9 września 1999r.,
- test wykrycia i analizy programów których

zachowanie zależy od daty (manipulowanie danymi, wywołanie procesów rezydentnych, porównanie danych, obsługa interfejsów),

— test środowiska pracy wybranej aplikacji, której zachowanie może zależeć od innych programów i wykorzystywanych narzędzi różnych producentów nie posiadających certyfikacji roku 2000.

Realizacja tych testów nie jest łatwa, zwłaszcza gdy nie istnieje kod badanych aplikacji. Posiłkować się zawsze należy odpowiednimi narzędziami testującymi, jak na przykład χSuds.

Tabela 6 Charakterystyka aplikacji związanych z problemem daty

Nazwa problemu	Szacowana liczba aplikacji z problemem daty [miliony]	Możliwość wykonania korekty na czas	Estymowana liczba aplikacji nieskorygowanych [miliony]	Okres uaktywniania się błędów
Rok 200	36	80%	7.2	1999 - 2001
Numery telefoniczne	25	855	3.75	2000 - 2025
Europieniądz	10	75%	2.5	1999 - 2005
Numery socjalne	15	90%	1.5	1050 - 2099
Unix	12	80%	1.2	2036 - 2038
Koniec zbioru	4	90%	0.4	1999 - 2001
Rok przestępny	2	90%	0.2	1999 - 2000
GPS	0.25	98%	0.05	1999 - 2000
Ogółem	104.25	87.25	16.755	1999 - 2099

Warto podkreślić, że analiza problemu Y2K spowodowała zwrócenie uwagi na innego typu anomalie możliwe do ujawnienia się w systemach informatycznych związanych z formatem daty. Listę takich problemów dotycząca USA przedstawia tabela 6 [12]. Problem numerów telefonicznych czy numerów socjalnych wynika z faktu przekroczenia możliwości ich zapamiętania w systemie z uwagi na niewystarczającą ilość zarezerwowanych na ten cel bitów. Problem europieniądza polega na braku symbolu (zwłaszcza w starszych systemach) oznaczającego pieniądz euro. Problem Unix wiąże się z wyzerowaniem rejestru odliczania czasu, posiadającego zakres od 1 stycznia 1970 do 19 stycznia 2038. Podobnie jak problem GPS (Global Possisioning System), dla którego wyzerowanie nastąpi już 21 sierpnia 1999 roku. Z tabeli 6 wynika, że większość wymienionych problemów nie zostanie rozwiązanych na czas, co oznacza jednak poważne kłopoty dla użytkowników tych systemów. Sądzę, że jest to również palący problem i w naszym kraju?

8. Uwagi końcowe

Z przedstawionych rozważań wynika, że problem oceny jakości (szacowanie, prognozowanie) jest rozwiązywany cząstkowo i wraz z rozwojem nowych typów aplikacji i nowych rodzajów systemów informatycznych (rozproszone, multimedialne, inteligentne) pozostaje nadal problemem otwartym. Problemy sterowania i zapewnienia jakości są jeszcze znacznie trudniejsze, gdyż wiele relacji pomiędzy charakterystykami jakości nie jest ściśle określonych (rozmyte), zaś kryterium optymalizacji jest wielokryterialne. Najbardziej obiecująca jest strategia krokowa, zbieżna do pewnego minimum lokalnego, wsparta odpowiednimi narzędziami umożliwiających duży zakres pomiarów i zapewniających prognozowanie jakości [14]. Narzędzia wspomagające projektowanie wykazują własności pakietów ekspertowych. Co więcej, każdy z projektów informatycznych posiada swoje unikalne cechy i wymaga indywidualnego podejścia, stąd zweryfikowanie modeli jakościowych jest sprawą trudną i wymaga przetworzenia ogromnej ilości odpowiedniej informacji, w tym często informacji historycznych, które z uwagi na tajemnicę firm są na ogół niedostępne.

Działalność ludzka jest bezpośrednio włączona w proces oceny i zapewnienia jakości. Zatem oprócz formalnych czy modelowych metod często istotne jest sformułowanie zdrowo-rozsądkowych zasad, których konsekwentne przestrzeganie prowadzi do poprawy jakości wytwarzanych systemów. Zatem opracowywane modele jakości powinny odzwierciedlać takie zasady, umożliwiać zarówno ocenę jakości jak i ukazywać zakres ich stosowalności. Obecnie buduje się wiele interesujących modeli pozwalających na badanie wybranych atrybutów czy charakterystyk jakościowych. Istnieje potrzeba zintegrowania ich w jedną metodologię oceny i zapewnienia jakości. Realizacja tego przedsięwzięcia wymaga jednak jeszcze wiele pracy, nie mniej postęp w tym kierunku jest zauważalny, czego przykładem może być choćby opracowanie interesujących metod pomiaru i wizualizacji jakości wykorzystywanych na różnych etapach życia systemu.

Dążenie do doskonałości jest cechą wrodzoną człowieka, stąd też nie rezygnuje on z budowy nowych kolejnych bardziej uniwersalnych modeli jakości, które stopniowo likwidować będą mniejsze i większe bariery utrudniające zastosowanie systematycznego podejścia. Rodzi się tylko pytanie, jakim kosztem to będzie realizowane i czy wystarczy człowiekowi determinacji do osiągnięcia powszechnie akceptowalnego rozwiązania.

Bibliografia

[1] T.Ball, S.G.Eick: Software Visualization in the Large. IEEE Computer, April, 1996, pp. 33-42.

[2] A.Behforooz, F.J.Hudson: Software Engineering Fundamentals, Oxford University Press, 1996.

[3] B.Begier: System jakości w tworzeniu oprogramowania. Materiały Konferencji Zintegrowane Systemy Informatyczne, Bydgoszcz, 1998, str. 101-11.

[4] B.W.Boehm: Software Enginnering Economics. Prentice-Hall, 1981.

[5] N.E.Fenton et al.: Software Metrics. A Rigorous & Practical Approach. Int. Thomson Computer Press, 1995.

[6] T.Gilb, D.Graham:Software Inspection. Addison Wesley, 1993.

[7] J.Herbsleb, D.Zubon, D.Goddenson et al.: Software Quality and the Capability Maturity Model. Communication of the ACM, vol. 40, June, 1997, pp.32-40.

[8] J.Górski, M.Jurkowlaniec, B.Nowicki: Doświadczenia w zakresie zastosowania metryk do oceny jakości oprogramowania. Informatyka 5, 1997.

[9] M.G.Hinchey,J.P.Bowen: Application of Formal Methods, Prentice-Hall, UK, 1995.

[10] W.Humphrey: Managing the Software Process. Addison Wesley, 1990.

[11] A.Jaszkiewicz: Inżynieria Oprogramowania. Helion, 1997.

[12] C.Jones: Bad days for software. IEEE Spectrum, Sept., 1998, pp. 47-52.

[13] C.Kaplan, R.Clark, V.Tang: Secrets of Software Quality. McGraw-Hill, Inc., 1995.

[14] H.Krawczyk, B.Wiszniewski: Analysis and testing of Distributed Software Applications. RSP, UK, 1998.

[15] M.Lorenz, J.Kidd: Object-Oriented Software Metrics. A Practical Guide. PTR Prentice Hall, 1994.

[16] S.Mizuro, Y.Akao: QFD: The Customer-Driven Approach to Quality Planning and Development. Asian Productivity Organization, Tokyo, 1994.

[17] M.C.Paulk, How ISO 9001 compares with the CMM. IEEE Software, Jan., 1995, pp.74-83.

[18] M.C.Paulk, B.Curtis, M.B.Chrissis et al.: Capability Maturity Model for Software. IEEE Software, Vol. 10, No. 4, July, 1993, pp. 18-27.

[19] Pressman RS: Software Engineering: A Practical Approach. McGraw-Hill, Inc., 1992.

Adam Czerminski

Zarządzanie przez jakość.
Rola testowania w zarządzaniu budową systemu w cyklu Architected RAD.

Cel

Chciałbym zastanowić się nad tym, czy testowanie ma być fazą w projekcie, w której jedynie stwierdza się stan oprogramowania (zły lub dobry) czy raczej powinno być ukonstytuowaną przez komórkę jakości projektu osią, wokół której obraca się cała działalność związana z wytworzeniem dobrej jakości kodu oprogramowania.

Jakość systemu

Na wstępie należy podkreślić, że interesująca będzie dla nas przede wszystkim jakość systemu (wykonywalnego kodu) - końcowego efektu cyklu wytwarzania oprogramowania, a nie jakość wszystkich elementów cyklu, nie mniej ważna, jednak nie będąca zasadniczym punktem rozważań.

To podkreślenie ma istotne znaczenie. Doświadczenie wielu lat inżynierii oprogramowania mówi, że to, z czym się użytkownik zmaga, to nie wymagania, nie ta, czy inna wersja projektu, ale żywy system, z którym pracuje. Z tego doświadczenia można wysnuć wniosek, że wytwarzanie kodu wymaga podejścia do jakości, które jest zdecydowanie ukierunkowane na efekt końcowy - działający system. Oznacza to, że takie podejście może się różnić od całościowego podejścia do jakości wszystkich elementów projektu.

Cykl Architected RAD

Cykl Architected RAD będący nowoczesnym cyklem wytwarzania oprogramowania będzie stanowił tło rozważań.

Cykl zakłada istnienie pewnej podstawy specyfikacyjnej np. specyfikacji wymagań kształtującej architekturę systemu w ten sposób, że jest możliwe określenie przyrostów sterujących budową systemu z wykorzystaniem narzędzi 4GL i metod prototypowania systemu wraz z użytkownikiem.

Postawienie problemu

Zarówno sformułowanie tematu jak i postawiony cel są na pierwszy rzut oka dość kontrowersyjne. Można odnieść wrażenie, że próbuje się umieścić testowanie w nieprawidłowym kontekście, że:

- miesza się testowanie, postrzegane na ogół jako jeden z końcowych etapów w projekcie, z zarządzaniem jakością, rozumianym jako zespół działań na poziomie strategicznym
- próbuje się nadać testowaniu nieproporcjonalne duże znaczenie projekcie w stosunku do programowania, czyniąc je być może czymś ważniejszym

Postawienie problemu nie jest jednak przypadkowe, gdyż jego celem jest skłonienie do refleksji i odpowiedzenie sobie na następujące pytanie:

Co w moich projektach jest głównym czynnikiem kreującym jakość wytwarzanego oprogramowania w fazie jego produkcji?

Niektóre z możliwych odpowiedzi:

- czas
- dostępne zasoby
- budżet
- skuteczne zarządzanie
- kompetencje ludzi
- metodyka

Na pierwszy rzut oka wydaje się, że testowanie nie jest nawet kandydatem na stopień kreatora jakości oprogramowania. Czy jednak na pewno?

Czynniki kształtujące jakość oprogramowania

Wymienione powyżej czynniki niewątpliwie mają wpływ na jakość kodu. Jaki jest jednak wpływ tych czynników na jakość? To pytanie będziemy analizować. Czy nie brakuje jednak tutaj jeszcze pewnego aspektu?

Jakość według norm ISO 9000 to: „ogół cech i właściwości wyrobu/produktu lub usługi decydujących o jego/jej zdolności do zaspokojenia stwierdzonych lub przewidywanych potrzeb". Potrzeby użytkownika są wyrażone w postaci wymagań, istnieją w architekturze systemu (np. w cyklu Architected RAD). Problem polega na tym, że trzeba je spełnić - muszą one być realizowane przez efekt końcowy w postaci działającego systemu. Tutaj rozpoczyna się ważne zadanie testowania.

Rola testowania

Przede wszystkim trzeba odpowiedzieć sobie na pytanie co to jest testowanie? Definicji testowania jest wiele. Przytoczę tutaj dwie główne, które wnoszą dwa istotne punkty widzenia:

„Testowanie jest procesem wykonywania programu lub systemu z intencją znalezienia błędu" Myers 1979

„Testowanie jest to jakakolwiek działalność mająca na celu ocenę (obliczenie, oszacowanie) atrybutów lub możliwości programu lub systemu i określenie, czy są one zgodne z oczekiwanymi wynikami" Hetzel 1979

Pierwsza definicja mówi, że testowanie powinno być skupione na poszukiwaniu błędów a nie na potwierdzaniu, tego, że program działa. Skuteczność testowania w zakresie wykrywania błędów zasadza się na tej definicji.

Druga definicja mówi, że testowanie ma za zadanie sprawdzenie, czy program działa zgodnie ze specyfikacją. Ta definicja z kolei zwraca uwagę, że testowanie nie powinno być nastawione na złamanie systemu, ale powinno doprowadzić do tego, że użytkownik otrzyma to co jest zgodne ze zgłoszonymi przez niego wymaganiami.

Te dwie definicje są konkurujące i do pewnego stopnia przeciwstawne, jednak w sumie dają istotne podwaliny pod określenie roli testowania.

Trzeba sobie uświadomić, że testowanie nie obejmuje tylko wykonania programu - np. zaprojektowanie i przygotowanie skutecznych testów jest ciężką pracą koncepcyjną i niewątpliwie wchodzi w zakres testowania. Można szerzej powiedzieć, że każda działalność nastawiona na niedopuszczenie do wystąpienia błędów w kodzie jak również działalność nastawiona na wykrycie błędów w kodzie wchodzi w zakres testowania.

W takim kontekście można zauważyć, że działania związane z testowaniem mają charakter pro-jakościowy.

Z drugiej strony z testowaniem wiąże się nieuchronnie wykrywanie błędów, zaś odpowiednie zarządzanie nimi może dać niezwykle wartościową informację o stanie jakości kodu.

W tym kontekście testowanie zaczyna nabierać charakteru instrumentu do zarządzania jakością kodu.

Wdrażanie zarządzania jakością kodu poprzez testowanie

Jeśli testowanie posiada znamiona instrumentu do zarządzania jakością kodu, wówczas trzeba odpowiedzieć sobie na pytanie jak wykorzystać testowanie do tego, aby kod był dobry. Poniżej podaję wybrane metody:

- wspólna specyfikacja kodu przez programistów i testerów

  programiści myślą jak zrobić, żeby działało, testerzy myślą o tym gdzie programiści nie pomyśleli, że nie będzie działać

- repozytorium testów

repozytorium gromadzi wiedzę na temat stanu jakości kodu w danym momencie, zarządzanie błędami zawartymi w repozytorium umożliwia śledzenie stanu jakości kodu

- tester jako wyspecjalizowany agent jakości patrzący na wszystkie fazy projektu z punktu widzenia efektu końcowego

  przewidywanie możliwych zagrożeń poprzez sprawdzenie jak przetestować dany fragment specyfikacji - niezależnie od poziomu jej szczegółowości - czy to będzie specyfikacja wymagań, czy szczegółowa specyfikacja funkcji w kodzie

Podsumowanie

Bliższe przyjrzenie się specyfice testowania pozwala dostrzec, że nie zawęża się ono jedynie do wykonania programu ale obejmuje szeroki zakres działań mających na celu z jednej strony zapobieganie powstawaniu błędom w kodzie i ich wykrywanie a z drugiej dostarczający wiedzy na temat stanu jakości kodu w danym momencie czasowym jak również na analizę jakości kodu na przestrzeni czasu.

Borys Stokalski

InfoViDE sp. z o.o., partner firm:
LBMS, SYBASE, SQA, Business Objects i SELECT
02-119 Warszawa, ul. Pruszkowska 17, tel. 659 04 84-6, faks 659 0343

Zasady Audytu Przedsięwzięć Informatycznych

Dokument zawiera obszerne fragmenty opracowania "Zasady i metodyka prowadzenia kontroli inwestycji informatycznych" stworzonego przez firmę InfoViDE na zamówienie Najwyższej Izby Kontroli.

1. Podstawa audytu - taksonomia zagrożeń

1.1. Ryzyko a audyt

Podstawą do właściwej organizacji audytu przedsięwzięć informatycznych jest dobre zrozumienie tych aspektów, które wiążą się z największymi zagrożeniami dla ich powodzenia, a które to aspekty są w ogromnej mierze pochodną niewłaściwych praktyk wytwórczych, niegospodarności i brakiem celowości.

Z drugiej strony ocena ryzyka w przedsięwzięciu może być sama w sobie jednym z podejmowanych działań kontrolnych. Osoba nie będąca specjalistą w dziedzinie informatyki może stosunkowo łatwo nauczyć się szacować ryzyko związane z prowadzeniem przedsięwzięcia informatycznego. Dzięki temu może wykrywać sytuacje, w których przedsięwzięcie zawiera obszary o bardzo dużym ryzyku, które nie zostały wystarczająco poważnie potraktowane przez wykonawców systemu i w związku z tym wymagają wnikliwszej kontroli.

1.2 Statystyczny obraz ryzyka w przedsięwzięciach informatycznych

Obserwując prawidłowości zachodzące w przedsięwzięciach informatycznych można stwierdzić, które czynniki w skali makro mają zasadniczy wpływ na osiągnięcie sukcesu lub poniesienie porażki.

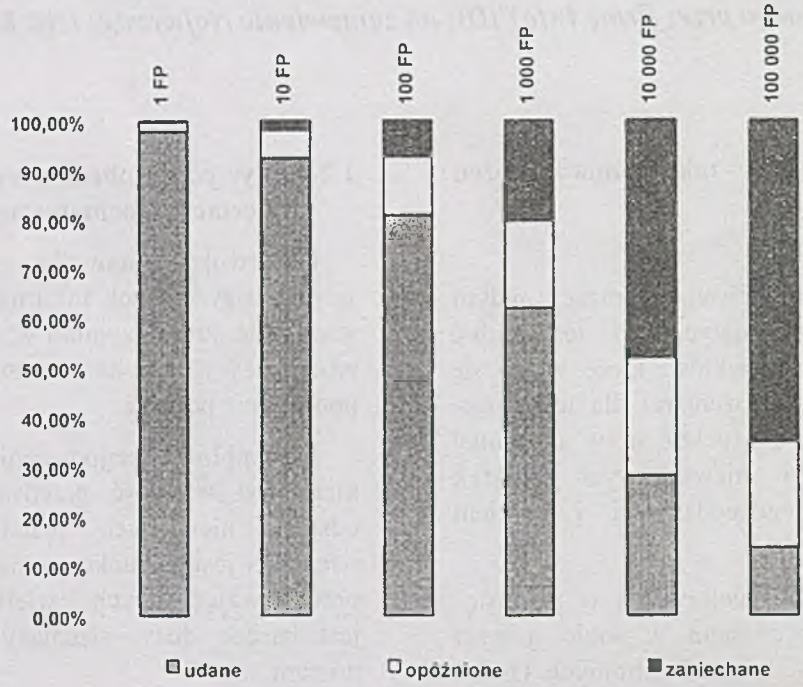
Niewątpliwie najpoważniejszym czynnikiem jest wielkość przedsięwzięcia. O ile odsetek nieudanych przedsięwzięć małej wielkości jest stosunkowo mały, o tyle wśród przedsięwzięć dużych i wielkich odsetek ten jest bardzo duży, sięgający kilkudziesięciu procent.

Poniższa tabela przedstawia rozkład prawdopodobieństwa zakończenia przedsięwzięcia przed terminem, w terminie, z opóźnieniem lub jego zaniechania w zależności od wielkości przedsięwzięcia. Wielkość przedsięwzięcia jest mierzona metodą punktów funkcyjnych (niezależna od sposobu realizacji przedsięwzięcia metoda pomiaru jego wielkości).

charakterystyka przedsięwzięcia	wielkość[1]	przed terminem	w terminie	opóźnione	zaniechane
małe programy	1 FP	14,68%	83,15%	1,92%	0,25%
aplikacje użytkownika	10 FP	11,08%	81,25%	5,67%	2,00%
duże aplikacje użytkownika	100 FP	6,06%	74,77%	11,83%	7,34%
programy tworzone przez firmy	**1 000 FP**	**1,24%**	**60,76%**	**17,67%**	**20,33%**
systemy informatyczne	**10 000 FP**	**0,14%**	**28,03%**	**23,83%**	**48,00%**
duże systemy informatyczne	**100 000 FP**	**0,00%**	**13,67%**	**21,33%**	**65,00%**

Bliższa analiza zamieszczonych poniżej danych wykazuje wyraźnie, że powyżej pewnej progowej wielkości przedsięwzięcia ryzyko nie udania przedsięwzięcia (opóźnienia lub zaniechania) wzrasta radykalnie. Można więc mówić o barierze udanego wykonywania przedsięwzięć na poziomie 1000 punktów funkcyjnych.

Znajomość rzędu wielkości przedsięwzięcia pozwala na podzielenie dużego przedsięwzięcia o dużym prawdopodobieństwie niepowodzenia na mniejsze, wydzielone przedsięwzięcia, oparte na wspólnej koncepcji określonej w czasie prac analitycznych, o znacznie mniejszym ryzyku niepowodzenia.



Szanse powodzenia przedsięwzięcia a jego wielkość.

[1] Dane do tabeli i wykresów w tym rozdziale zostały zaczerpnięte z książki — *Capers Jones, Patterns of Software Systems Failure and Success, International Thomson Computer Press* 1996, w której przeanalizowano 6 700 przedsięwzięć informatycznych realizowanych na całym świecie.

Innym ważnym czynnikiem wpływającym na prawdopodobieństwo sukcesu lub porażki jest doświadczenie zespołu realizującego prace: kierowników, wykonawców i ekspertów. Można ten czynnik uogólnić nazywając go „socjologią" przedsięwzięcia, czyli strukturą doświadczeń zespołu związanych z konkretną sferą informatyzacji. Również technologia budowy systemów nie jest bez wpływu na szansę powodzenia projektu, choć jej wpływ polega głównie na tym, że musi ona być właściwie dobrana do specyfiki rozwiązywanego problemu.

Prawdopodobnie najważniejszym elementem wpływającym na to, czy uda się zrealizować przedsięwzięcie na czas i w założonym budżecie, jest ogólnie rozumiana kultura zarządzania. Przejawia się ona w stosowaniu odpowiednich metod zarządzania, jak również w stosowaniu właściwych narzędzi wspomagających zarządzanie. Czynnik ten obejmuje również zagadnienia zapewnienia i kontroli jakości.

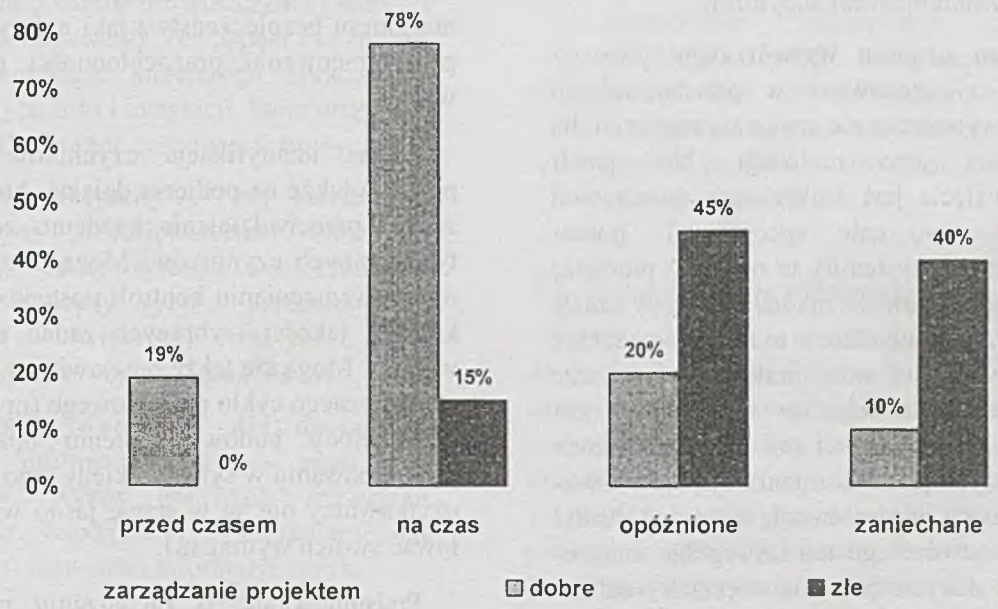
Czynniki związane z zarządzaniem przedsięwzięciem

— szacowanie pracochłonności przedsięwzięcia (ręczne albo automatyczne),

— planowanie (ręczne albo automatyczne),

— śledzenie stopnia zaawansowania prac (nieformalne albo formalne),

— określanie nakładów na kontrolę jakości (minimalne albo optymalne).



Praktyki związane z zarządzaniem przedsięwzięciem, a szanse na powodzenia przedsięwzięcia

O każdym przedsięwzięciu informatycznym można powiedzieć, iż może się nie udać z wielu powodów związanych z „zarządzaniem":

— przedsięwzięcie jest zaplanowane źle lub wcale, nie znane są zatem koszty, ryzyko i korzyści płynące z inwestycji,

— postęp prac jest niewłaściwy lub też nie jest na bieżąco oceniany z powodu braku odpowiednich procedur kontroli postępu prac i planu, względem którego taki postęp prac mógłby być określany,

— jakość produktów końcowych nie jest właściwa, ponieważ nie określono sposobów zapewnienia jakości oraz kryteriów i procedur kontroli jakości,

— zakres prac jest wielokrotnie poszerzany lub zawężany w sposób nie kontrolowany przez kadrę zarządzającą przedsięwzięciem oraz bez niezbędnej w tym

wypadku analizy skutków zmiany zakresu dla przedsięwzięcia (to zjawisko określane bywa mianem „pełzania" lub „płynności" zakresu).

W związku z tym warto myśleć o zadaniach audytorów w następujących aspektach:

Analiza stanu przedsięwzięcia, która pozwala wcześnie zidentyfikować zagrożenia dla realizacji celów dużych wieloletnich przedsięwzięć informatycznych. Wczesne wykrycie zagrożeń jest istotne dla szybkiego podjęcia działań umożliwiających uniknięcia strat wynikających z niegospodarności lub załamania przedsięwzięcia.

Analiza celowości jest niezbędna do oceny trafności decyzji o podjęciu i zakresie prac. Powinna być prowadzona dla wszystkich przedsięwzięć informatycznych będących w sferze zainteresowań audytorów.

Analiza procesu wytwórczego, pozwala oceniać czy stosowane w przedsięwzięciu praktyki wytwórcze nie stwarzają zagrożeń dla powodzenia jego realizacji. Np. jeżeli przedsięwzięcie jest realizowane dwufazowo (najpierw stworzenie specyfikacji, potem implementacja systemu), to mimo iż pierwsza część przedsięwzięcia zakończyła się w czasie i przewidzianym budżecie to może się zdarzyć że stosowane w nim praktyki wytwórcze doprowadziły do tego że specyfikacja jest nieodpowiedniej jakości (np. nie przedstawia rzeczywistych potrzeb organizacji, nie pozwala na sprawną implementację systemu). Audyt procesu wytwórczego ma szczególne znaczenie przy dużych przedsięwzięciach realizowanych przez wielu poddostawców.

Wszystkie metody zarządzania przedsięwzięciami mają wspólne korzenie i zazwyczaj można doszukać się w nich procedur planowania, szacowania czasochłonności i czasu realizacji, budżetowania, kontroli postępu prac, jakości, zmian, itp. Mówiąc inaczej: procedur przeciwdziałania typowym czynnikom ryzyka przedsięwzięć informatycznych.

Dodatkowo gdy, realizowane jest konkretne przedsięwzięcie niezbędne jest narzędzie, które może służyć bardziej szczegółowej ocenie ryzyka tego przedsięwzięcia. Dysponując takim narzędziem można zidentyfikować czynniki ryzyka, a następnie zaplanować przedsięwzięcie w taki sposób, aby przeciwdziałać zagrożeniom lub mieć „awaryjne" scenariusze postępowania na wypadek pojawienia się przeanalizowanych wcześniej „nieprzewidzianych" trudności. Narzędziem takim jest metryka ryzyka.

1.3. Metryka ryzyka

Metryka pozwala ona na systematyczny przegląd kilkudziesięciu typowych czynników ryzyka i przypisanie każdemu z nich odpowiedniej miary. Zsumowanie miar przypisanych poszczególnym czynnikom ryzyka pozwala na klasyfikację ryzyka, np. określenie, czy ryzyko przedsięwzięcia jest „małe", „średnie", „duże" lub „ekstremalnie wysokie". Tak sklasyfikowane ryzyko pozwala na określenie wielkości marginesu bezpieczeństwa jaki należy przyjąć przy szacowaniu pracochłonności przedsięwzięcia.

Jawna identyfikacja czynników ryzyka pozwala także na podjęcie działań, które mają za cel przeciwdziałanie każdemu ze zidentyfikowanych czynników. Mogą one polegać np. na wzmacnianiu kontroli postępu prac lub kontroli jakości wybranych zadań przedsięwzięcia. Mogą się także przejawiać w doborze specyficznego cyklu projektowego (np. wybranie metody budowy systemu opartej na prototypowaniu w sytuacji, kiedy jego przyszli użytkownicy nie są w stanie jasno wyartykułować swoich wymagań).

Podsumowując: o zarządzaniu ryzykiem można mówić biorąc pod uwagę kilka poziomów działań związanych z inwestycjami informatycznymi:

Planowanie strategiczne — wtedy, kiedy planowane są działania w skali przedsiębiorstwa zmierzające do implementacji pewnej liczby systemów informatycznych dostosowanych do strategii jego działania (*Strategic Planning*). W przypadku, gdy nie jest możliwe sformułowanie strategii, rozsądną taktyką informatyzacji firmy jest kolejna realizacja przedsięwzięć informatycznych o „lokalnie" najwyższych zwrotach inwestycji przy minimalnym ryzyku i czasie realizacji.

Generalnie rzecz biorąc ocena opłacalności inwestycji informatycznych nie powinna się różnić od oceny opłacalności innych inwestycji. Specyficznym dla przedsięwzięć informatycznych problemem jest sposób kwantyfikacji korzyści niematerialnych płynących z wprowadzenia systemu informatycznego (np. zwiększenie zdrowotności obywateli). Innym specyficznym czynnikiem jest określanie kosztów utrzymania systemu związane nie tylko z kosztem pracy ludzkiej, ale także związanej z modernizacją infrastruktury technicznej (sprzęt komputerowy, oprogramowanie, sieć komputerowa).

Kontrolowanie procesu wytwórczego — poprzez przyjęcie właściwych metod budowania systemów informatycznych i wdrożenie praktyk ulepszania tych metod w sposób nadążający za zmianami technologii i w oparciu o własne doświadczenia (*Software Process Management*). Ten aspekt ma znaczenie w kontekście niniejszego opracowania tylko dla tych firm i instytucji, które utrzymują własne rozbudowane zespoły informatyczne.

Szczebel taktyczny — gdy zarządzamy pojedynczym przedsięwzięciem informatycznym. W takiej sytuacji konieczna jest umiejętność oceny ryzyka przedsięwzięcia i przyjęcie właściwej taktyki walki z zagrożeniami (*Project Management*).

Określanie warunków dla dostawców systemów informatycznych — organizacja przetargów, wybór gotowych rozwiązań, organizacja współpracy z oferentami lub dostawcami rozwiązań informatycznych.

2. Dobre praktyki - podstawa prac audytorskich

Brak któregokolwiek z omówionych poniżej elementów dobrej praktyki informatycznej można uznać za poważny sygnał, że prace nad systemem mogą być prowadzone niezgodnie z zasadami sztuki i w konsekwencji może to doprowadzić do poważnych perturbacji z planowym zakończeniem przedsięwzięcia oraz jakością dostarczanych produktów. Problemy z terminowym zakończeniem przedsięwzięcia mogą przynieść następujące niekorzystne skutki:

— niemożność osiągnięcia przez przedsięwzięcie założonych celów (np. nie wykonanie na czas systemu obsługi wyborów parlamentarnych),

— powoduje koszty związane z odsunięciem w czasie korzyści spodziewanych po wprowadzeniu systemu (np. późniejsze wdrożenie mechanizmów zmniejszających koszty działania instytucji),

— przedłużanie się czasu realizacji przedsięwzięcia może wiązać się również z dodatkowym wydatkowaniem środków co niekorzystnie wpływa na opłacalność przedsięwzięcia,

— jeżeli w przedsięwzięciu terminowa realizacja poszczególnych etapów jest związana z logistyką dostaw (np. dostarczanie sprzętu komputerowego) może powodować niekorzystne zjawisko w którym sprzęt komputerowy nie może być wykorzystany zgodnie z przeznaczeniem, oraz następuje jego szybka techniczna dewaluacja i upływ praw gwarancyjnych.

2.1. Zarządzanie przedsięwzięciami

Jest to obszar, w którym prawidłowe działanie daje największe możliwości ograniczenia ryzyka niepowodzenia przedsięwzięcia. Zarządzanie przedsięwzięciem pozwala prawidłowo reagować na sytuacje kryzysowe i ograniczać straty powstające w ich wyniku (np. przekroczenie terminów, budżetu, niemożność zrealizowania założonego zakresu prac, nietrafny wybór architektury technicznej systemu).

Dokumentacja związana z zarządzaniem przedsięwzięciem powinna zawierać:

— opis organizacji zespołu i prac projektowych.

— definicję zakresu przedsięwzięcia,

— plan projektu, oraz plan zapewnienia jakości,

— procedury kontroli i raportowania przebiegu prac projektowych,

— ocenę ryzyka.

Bez powyższych elementów nie można mówić o dobrym zarządzaniu przedsięwzięciem w sensie możliwości kontroli stanu prac, harmonogramu, budżetu i jakości.

2.1.1 Organizacja zespołu i prac projektowych

Czynnikiem decydującym o skutecznym prowadzeniu prac projektowych, właściwym określeniu i zrozumieniu zakresu przedsięwzięcia oraz o powstaniu realistycznego planu działań jest współpraca zamawiającego i wykonawcy w ramach właściwie ustalonej struktury organizacyjnej przedsięwzięcia.

Przez właściwie ustaloną strukturę przedsięwzięcia rozumiemy:

— strukturę w której kompetencje są rozłożone odpowiednio do wagi podejmowanych decyzji (np. kierownik projektu powinien mieć wszystkie kompetencje niezbędne do wyznaczania i egzekwowania wykonania zadań na poziomie operacyjnym, niezależnie od tego czy są to zadania przydzielane pracownikom wykonawcy, czy zamawiającego, a zarząd projektu powinien mieć kompetencje pozwalające podejmować decyzje strategiczne dla przedsięwzięcia, takie jak: budżet, zakres prac, zmiany zakresu),

— jest jasno określony zakres odpowiedzialności uczestników przedsięwzięcia, który jest adekwatny do ich kompetencji,

— określone są ścieżki i procedury raportowania postępu prac i zagrożeń.

Opis struktury organizacyjnej przedsięwzięcia powinien określać:

— obsadę stanowisk niezbędnych dla przedsięwzięcia: zarząd projektu, kierownik projektu (etapu), zespołów projektowych, grupy kontroli jakości, przedstawicieli nabywcy współpracujących z zespołami projektowymi,

— hierarchię zależności między uczestnikami przedsięwzięcia,

— zakres odpowiedzialności wszystkich uczestników przedsięwzięcia,

— sposób raportowania postępu prac i sygnalizowania sytuacji krytycznych.

Za poważne błędy w organizowaniu prac nad przedsięwzięciem należy uznać:

— brak jasno określonych zakresów odpowiedzialności uczestników przedsięwzięcia,

— rozmycie odpowiedzialności za podejmowane decyzje przy braku jasno określonych centrów decyzyjnych na poziomie operacyjnym i strategicznym,

— brak jasno określonych ścieżek przepływu informacji w strukturze organizacyjnej przedsięwzięcia.

2.1.2 Planowanie projektu

Najczęściej celem przedsięwzięcia informatycznego jest wytworzenie pewnych rozwiązań z wykorzystaniem dostępnej technologii. Z tego względu dobrze wykonany plan powinien koncentrować się na produktach (np. dokumentacja, oprogramowanie, przeszkolony zespół), które tworzone są w czasie trwania przedsięwzięcia.

Planowanie w przedsięwzięciu informatycznym polega na:

— określeniu zakresu prac,

— identyfikacji i zdefiniowaniu wszystkich oczekiwanych produktów końcowych,

— zaproponowaniu „ciągu technologicznego" dla kluczowych produktów pośrednich wraz z określeniem standardów ich wykonania oraz z uwzględnieniem możliwości i ograniczeń technologicznych,

— określeniu sposobu wykonania produktów pośrednich — technik, narzędzi, kwalifikacji personelu i wynikających stąd zadań,

— optymalnym rozplanowaniu realizacji zadań z uwzględnieniem dostępnych zasobów, ograniczeń i czynników ryzyka.

Plan projektu powinien zawierać:

— krótki opis przyjętego sposobu prowadzenia prac w przedsięwzięciu wraz z określeniem podziału prac na etapy,

— harmonogramy prac, w których uwzględnione są zadania jakie trzeba wykonać na danym etapie prac wraz z zależnościami między nimi, produkty końcowe powstające w projekcie, zależności zewnętrzne jakim podlegają prace projektowe oraz zużycie zasobów niezbędnych do wykonania zaplanowanych zadań.

Techniki związane z przygotowaniem planu i harmonogramu takie jak analiza ścieżki krytycznej, bilansowanie obciążeń zasobów są elementem większości poważnych metodyk zażądania projektami i nie są charakterystyczne dla przedsięwzięć informatycznych. Większość dużych dostawców usług integratorskich posiada własne metody zarządzania projektami oparte o ogólnie przyjęte standardy zarządzania. Ważne jest by stosujące je w praktyce osoby posiadały zarówno formalne wykształcenie jak i niezbędną dozę wiedzy z zastosowania tego typu metod.

Plan projektu konstruowany jest zwykle w sposób nadążny, tzn.:

— w trakcie inicjacji projektu określony jest ramowy plan całego przedsięwzięcia,

— przed rozpoczęciem prac etapu tworzony jest szczegółowy plan tego etapu,

— po zakończeniu prac etapu ramowy plan całego przedsięwzięcia może zostać zmodyfikowany w związku z okolicznościami wynikłymi w trakcie realizacji etapu (np. zmiana zakresu prac, przyjęcie innej niż przewidywano architektury systemu).

Oprócz powyższych elementów wykonawca powinien szczegółowo zdefiniować produkty przedsięwzięcia wskazując standardy związane z wykorzystywanym procesem wytwórczym stanowiące podstawę jego działań oraz opisując podstawowe adaptacje tych standardów zastosowane w przedsięwzięciu (np. opisać, które elementy metodyki są pomijane i dlaczego). Dokumentacja tego typu powinna zawierać:

— listę produktów niezbędnych do osiągnięcia celu z określeniem zależności między nimi,

— opis technik wykorzystywanych w czasie tworzenia poszczególnych produktów,

— opis narzędzi wspomagających korzystanie ze stosowanych technik,

— kryteria i procedury pozwalające śledzić jakość powstającego produktu oraz podejmować stosowne działania zaradcze i pro jakościowe,

— określenie poziomu umiejętności i kwalifikacji personelu jakie są niezbędne do osiągnięcia celu,

— opis metryk (kryteriów odbioru) pozwalających na ocenianie jakości produktu oraz szacowanie kosztów jego stworzenia.

Przy pomocy dokumentacji definiującej produkty przedsięwzięcia możliwe jest rzetelne szacowanie czasu, zasobów i kosztów oraz uzgodnienie wymagań dotyczących produktów między wykonawcą a odbiorcą. Brak takiej dokumentacji jest sygnałem, że nie wiadomo dokładnie co ma być wykonane w przedsięwzięciu oraz jakie będą kryteria odbioru. Może to doprowadzić do sytuacji, w której odbiorca stwierdza, że wykonany produkt nie jest tym, który zamawiał, nie mając jednocześnie możliwości egzekwowania swoich roszczeń.

2.1.3 Plan zapewnienia jakości

Plan jakości powinien określać sposób w jaki dostawca zamierza zapewnić jakość powstających produktów, uwzględniając nie tylko kontrolę gotowego produktu, ale także działania zapobiegające powstawaniu błędów w tracie prac nad nim.

Plan zapewnienia jakości jest podstawą do podejmowania działań kontrolnych w ramach przedsięwzięcia oraz powinien służyć jako podstawowy materiał do przygotowania niezależnego audytu zewnętrznego przedsięwzięcia.

W planie zapewnienia jakości powinny się znaleźć:

- określenie zakresu odpowiedzialności za jakość prac w zespołach projektowych,
- określenie działań pro jakościowych podejmowanych w czasie prac,
- definicje produktów powstających w projekcie wraz z kryteriami ich odbioru (w przypadku produktów standardowych wystarczy odwołanie się do metodyki, a w przypadku produktów niestandardowych niezbędny jest szczegółowy opis), przy czym zdefiniowanie produktów można przesunąć do czasu planowania szczegółowego harmonogramu etapu w czasie którego mają powstać,
- odniesienia do standardów, metodyk i notacji wykorzystywanych w projekcie,
- opis organizacji zespołu kontroli jakości i procedur jego działania,
- harmonogram kontroli jakości wkomponowany w harmonogram przedsięwzięcia uwzględniający testy jednostkowe, funkcjonalne, niefunkcjonalne i integracyjne,
- plan testów akceptacyjnych przy odbiorze systemu.

Planowana kontrola jakości powinna obejmować następujące zadania:

- identyfikację produktów podlegających kontroli,
- identyfikację celów i zakresu kontroli,
- identyfikację atrybutów jakości,
- określenie formy kontroli (przegląd strukturalny, przegląd wzajemny, inspekcja).

Brak planu zapewnienia jakości może w projekcie pociągać za sobą następujące skutki:

- brak możliwości obiektywnej oceny działań podjętych w przedsięwzięciu w celu zapewnienia jakości,
- niemożność oszacowania nakładów niezbędnych do utrzymania jakości produktów na wymaganym poziomie,
- kontrole jakości wykonywane są „w miarę potrzeb" i ich efektywność jest w takich przypadkach problematyczna,
- niejasne kryteria oceny produktów uniemożliwiają obiektywną ocenę ich jakości.

2.1.4. Procedury kontroli przebiegu prac projektowych

Kontrola przedsięwzięcia jest zadaniem należącym do obowiązków Kierownika i Zarządu projektu.

Należy pamiętać, że rutynowa kontrola przebiegu przedsięwzięcia, jakości powstających produktów, zużycia zasobów i stopnia realizacji postawionych zadań nie jest działaniem, które można realizować *ad hoc*, lecz wymaga podjęcia odpowiednich czynności organizacyjnych i określenia procedur pracy.

Jasno określone procedury pracy należy zdefiniować również dla takich działań jak:

- zarządzanie zmianami, które wymaga podejmowania trudnych decyzji o daleko idących skutkach dla całego przedsięwzięcia, gdzie dodatkowo ścierają kompetencje i interesy zamawiającego i dostawcy. Wymagania określane dla dużych przedsięwzięć informatycznych mają tendencję do zmieniania się w trakcie trwania przedsięwzięcia, natomiast dostawca jest ograniczony umową i budżetem, nie może zgadzać się na dowolne zmiany wymagań stawianych przez zamawiającego. W takiej sytuacji tylko sformalizowana procedura zgłaszania zmian, analizy skutków i kosztów oraz akceptowania zmian zakresu przedsięwzięcia pozwala na sprawne zarządzanie tą sferą, inne metody wprowadzania zmian powinny budzić zastrzeżenia: zwykle jest to albo jest to arbitralna decyzja zamawiającego, która może naruszyć możliwości wykonawcze zespołu, albo arbitralna decyzja dostawcy.
- odbiór prac, który wymaga określenia terminów odbioru, procedur, przyczyn

odrzucenia produktu, sposobu wykorzystania zewnętrznych audytorów.

Do zadań Kierownika projektu związanych z kontrolą przedsięwzięcia, należą:

— śledzenie czasu i stopnia realizacji zadań oraz wykonywanych produktów,

— sporządzanie co 1–2 tygodnie „Raportów z postępu prac" przeznaczonych dla Zarządu projektu,

— organizowanie spotkań kontrolnych i przeglądów zespołowych z udziałem poszczególnych członków Zespołu wykonawczego,

— reagowanie na sytuacje kryzysowe zagrażające terminowemu wykonaniu prac,

— analizowanie skutków zmian w zakresie przedsięwzięcia i korygowanie harmonogramu i budżetu przedsięwzięcia.

Do zadań Zarządu projektu związanych z kontrolą przedsięwzięcia, należą:

— analizowanie otrzymanych od Kierownika projektu „Raportów z postępu prac".

— akceptowanie lub odrzucanie raportów Kierownika projektu i gotowych produktów.

Postęp prac powinien być mierzony przez odniesienie do aktualnego harmonogramu prac. Stopień zaawansowania wykonania czynności, etapów i całego przedsięwzięcia powinien być określany i rejestrowany w „Raportach z postępu prac".

Brak dokumentacji ukazującej systematyczne kontrolowanie przebiegu prac przez Kierownika i Zarząd projektu należy uznać za bardzo poważny błąd w sztuce. Nawet jeżeli prowadzący przedsięwzięcie orientują się na bieżąco co robią ich podwładni, to bez formalnego raportowania nie można ustalić zakresu odpowiedzialności za ewentualne opóźnienia i przekroczenie budżetu.

2.1.5. Ocena ryzyka

Potrzeba oceny ryzyka w przedsięwzięciu informatycznym wynika stąd, iż nie można zakładać z całkowitą pewnością, że osiągnięte zostaną planowane korzyści w zaplanowanym czasie i budżecie.

Jeżeli przedsięwzięcie będzie przebiegało sprawniej niż zakładano to nie jest to groźne. Należy jednak w takim przypadku rozważyć zasadność poczynionych wstępnych założeń i szacowań oraz czy nie została zawyżona kwota ustalona jako zapłata za wykonywaną pracę.

Bardziej niebezpieczna (i znacznie częściej spotykana) jest sytuacja, gdy prace przebiegają znacznie wolniej (i drożej) niż początkowo planowano. Opóźnienia prac wobec przyjętego harmonogramu mogą wynikać z następujących powodów:

— optymistyczne szacowanie zostało wymuszone w trakcie negocjacji kontraktu przez Zamawiającego lub z chęci „przebicia cenowego" konkurentów przez Dostawcę,

— nie zachowana została należyta staranność przy wykonywaniu szacowania pracochłonności przedsięwzięcia.

Przy szacowaniu pracochłonności można zastosować następujące metody:

— wyznaczenie sztywnych ram czasowych na realizację poszczególnych etapów przedsięwzięcia, kosztem ewentualnego ograniczenia zakresu prac — daje to zamawiającemu pewność co do poziomu kosztów przedsięwzięcia, natomiast problemem jest uzyskanie zadawalającej funkcjonalności systemu,

— szacowanie formalne prac (np. przy wykorzystaniu metody punktów funkcyjnych) — jest ono charakterystyczne dla prac prowadzonych z rozłożeniem na fazy: umowa wstępna, szczegółowa analiza systemu i renegocjowanie umowy na podstawie szczegółowych szacunków dla poszczególnych części systemu.

Zarząd projektu powinien wiedzieć jakie mogą być zagrożenia dla powodzenia przedsięwzięcia i jakie konsekwencje (np. finansowe) wiążą się z nimi. Poniżej została przedstawiona lista typowych zagrożeń dla przedsięwzięć informatycznych, które mają znaczący wpływ na powodzenie i koszty przedsięwzięcia:

- niedostateczne kwalifikacje i doświadczenie kadry kierowniczej projektu,
- niedostateczne kwalifikacje zespołu projektowego,
- problemy z technologią wybraną do realizacji przedsięwzięcia, które zwykle pojawiają się, gdy stosowane są nowe i nie sprawdzone na rynku rozwiązania,
- zmienne warunki ekonomiczne lub prawne gwałtownie zmieniające wymagania wobec systemu (np. zmiana ustawy o rachunkowości zmienia wymagania stawiane przed systemem księgowym).

Dodatkowo dobrze jest przeanalizować metody unikania lub minimalizowania strat związanych z wystąpieniem każdego ze zidentyfikowanych zagrożeń. Przykładami działań zapobiegawczych mogą być:

- stosowanie pilotażu technicznego w celu przetestowania wybranej dla przedsięwzięcia technologii,
- analiza tendencji gospodarczych i legislacyjnych,
- szkolenia zespołu podnoszące jego kwalifikacje, a co za tym idzie zmniejszające ryzyko popełnienia błędu,
- wykorzystanie ekspertów technicznych i problemowych.

2.2. Dokumentacja

Do prowadzenia i oceny prawidłowego przebiegu przedsięwzięcia niezbędne są następujące rodzaje dokumentacji:

- dokumentacja określająca cel i zakres przedsięwzięcia a także wymagania stawiane budowanemu systemowi informatycznemu,
- dokumentacja zarządcza — na którą powinny składać się co najmniej: struktura organizacyjna przedsięwzięcia, aktualizowany plan projektu oraz harmonogram i raporty z postępu prac,
- dokumentacja projektowa i techniczna w pełni opisująca przyjęte w projekcie rozwiązania,
- dokumentacja zawierająca opisy i wyniki testów,
- dokumentacja dla użytkowników systemu umożliwiająca im pełne wykorzystanie funkcji systemu oraz utrzymanie systemu w „ruchu".

Dokumentacja jest podstawą do oceny wszystkich aspektów przedsięwzięcia informatycznego. Z tego powodu ważne jest zwracanie uwagi w jaki sposób jest traktowane w projekcie prowadzenie pełnej dokumentacji projektowej.

Aby obiektywnie ocenić jakość dokumentu niezbędne jest określenie, przed rozpoczęciem prac nad nim, zasad na podstawie których dokonywana będzie ocena. Brak definicji kryteriów oceny produktów (dokumentacji) należy uznać za poważne uchybienie w procesie wytwórczym. Kwestie związane z kryteriami oceny produktów opisano w części poświęconej kontroli jakości.

2.3. Kontrola wersji

Przez kontrolę wersji rozumiemy zestaw działań, które pozwalają nadzorować cykl tworzenia i modyfikacji wszystkich produktów powstających w czasie prac nad systemem.

W ramach kontroli wersji rozróżniamy następujące działania:

- jednoznaczna identyfikacja wszystkich produktów powstających w czasie prac projektowych, dotyczy to zarówno dokumentacji papierowej, jak i produktów dostępnych w wersji elektronicznej (np. kody źródłowe oprogramowania),
- potwierdzanie zasadności zmian i tworzenie nowej wersji produktu oraz przeprowadzenie analizy wpływu zmian na inne produkty,
- nadawanie oznaczeń kolejnym wersjom produktów, tak aby pozwalały na

identyfikację kolejnych wydań (wersji) i aby w każdej chwili trwania przedsięwzięcia można było stwierdzić, która wersja danego produktu jest wersją aktualną (obowiązującą),

— stworzenie systemu rozpowszechniania wśród uczestników przedsięwzięcia aktualnej wersji produktów, tak aby prowadzone prace odnosiły się zawsze do aktualnych wersji,

— uaktualnianie odniesień do zaktualizowanego produktu, tak aby dotyczyły zawsze aktualnej wersji,

— wprowadzanie zamian w produktach powiązanych z aktualizowanym produktem.

Brak tego typu mechanizmów może doprowadzić do katastrofy mimo, iż całość prac była realizowana zgodnie z zasadami sztuki. Problem ten nabiera szczególnego znaczenia w następujących przypadkach:

— zmiany zakresu przedsięwzięcia, a co za tym idzie modyfikacje dokumentacji projektowej powstałej do tej pory; nie panowanie nad aktualizowaniem dokumentacji może doprowadzić do sytuacji, gdy w pracach wykorzystywane są dokumenty dotyczące różnych zakresów prac.

— testowanie i wdrażanie systemu, kiedy to poprawki nanoszone są bardzo szybko, a w takich przypadkach bardzo często dochodzi do sytuacji, w której kontrolowanie zmian (wersji) schodzi na plan dalszy,

— kontrolowanie w jaki sposób zmiany w jednym dokumencie wpływają na aktualność (poprawność) innych dokumentów: w prostym przypadku może to być tylko zmiana numerów wersji w miejscach gdzie są odwołania do modyfikowanego dokumentu, w skomplikowanym, modyfikacja jednego dokumentu wymusza duże zmiany w innych dokumentach.

— dokumentowanie zmian powstających w czasie wdrażania i eksploatacji systemu, co pozwala na ich kontrolowanie i ewentualny powrót do starej wersji w przypadku niezadowalającej jakości nowej wersji.

2.4 Planowanie i kontrola jakości

Dobrze zaplanowane działania mające na celu zapewnienie wysokiej jakości systemu informatycznego są niezbędnym składnikiem zarządzania całego przedsięwzięcia. Kontrola jakości w przedsięwzięciu nie może sprowadzać się tylko do końcowego odbioru gotowego oprogramowania, istotne jest aby cała gama działań zapobiegawczo–kontrolnych była zapisana w planie projektu od jego inicjacji do wdrożenia. Tylko tak zaplanowane działania mają szansę na szybkie wykrywanie nieprawidłowości w projekcie i co za tym idzie minimalizowanie ewentualnych strat i opóźnień.

Kontrola jest efektywna szczególnie wtedy gdy:

— niezbędne jest formalne przekazanie produktu do kontynuowania dalszych prac, co stanowi podstawę do rozliczeń wykonawcy,

— koszty wykrycia i/lub naprawy błędów w dalszych etapach projektu są znacznie wyższe niż koszty wykrycia i naprawy błędów w bieżącym etapie projektu,

— jakość produktów obecnego etapu jest krytyczna dla jakości produktów wytwarzanych na dalszych etapach projektu,

— skutki ewentualnych błędów są bardzo poważne.

2.4.1 Zapewnianie jakości

Zapewnianie jakości to zbiór czynności, których efektem ma być powstanie produktu o założonej jakości.

Możemy rozróżnić trzy poziomy zapewniania jakości:

1. nadzorowanie produktu — badanie jakości produktu końcowego,
2. nadzorowanie procesu — monitorowanie poziomu usterek, aby określić te elementy procesu w których popełniane są błędy,
3. ulepszanie procesu — tworzenie takiego projektu i procesu produkcji, aby zminimalizować liczbę ewentualnych błędów.

Obecnie zapewnienie jakości oprogramowania znajduje się na poziomie (1) i polega głównie na organizowaniu inspekcji i przeglądów, których celem jest ocena gotowego produktu. Osiągniecie poziomów (2) i (3) wymaga nie tylko odpowiedniej organizacji prac, ale zależy także od wskaźników pomiarowych i sposobu monitorowania jakości. Polepszanie projektu produktu wymaga dokładnego przewidywania działań niezbędnych w czasie tworzenia i działania.

Przyjęte jest, że po zakończeniu każdego etapu prac następuje formalny odbiór produktów, które powstały w jego trakcie (i zostały „zaplanowane" jako produkty końcowe w planie etapu). Odbiór taki powinien zakładać co najmniej ocenę produktu ze strony przedstawicieli Zamawiającego, a w kluczowych dla przedsięwzięcia momentach zasadne jest powołanie zewnętrznej grupy oceniającej.

Nie można jednak sprowadzić zapewniania jakości tylko do formalnych kontroli. Takie postawienie sprawy przez dostawcę świadczy o niezrozumieniu problemu jakości i nieefektywności jego działań. Działania prewencyjne powinny być jak lepiej wkomponowane w metodykę prowadzonych prac i nakłady poniesione na działania pro jakościowe wielokrotnie się zwracają ponieważ powstają produkty lepsze, a ewentualne błędy wykrywane są bardzo szybko, często jeszcze przed formalną kontrolą jakości. Formalne kontrole powinny być tylko potwierdzeniem tego jak dobre są stworzone produkty.

2.5 Standardy pracy zespołu

Zdefiniowane i dostępne standardy pracy zespołu pomagają nie tylko w prowadzeniu prac nad systemem, ale także są czynnikiem znacznie ułatwiającym i obiektywizującym ocenę tych prac.

Do istotnych standardów pracy zespołu projektowego można zaliczyć:

- opis architektury technicznej systemu —zawierający opis elementów systemu komputerowego, topologię sieci, odwołanie do norm i standardów przemysłowych przyjętych w projekcie (np. telekomunikacyjnych, baz danych),
- standardów komunikacji z użytkownikiem (co jest szczególnie istotne przy konstruowaniu aplikacji z graficznym interfejsem użytkownika np. przy wykorzystaniu *Windows*), zachowanie jednolitych i przyjętych standardów interakcji (szczególnie dla aplikacji biurowych) znacznie ułatwia wdrożenie i szkolenie użytkowników, dobrą praktyką jest przed rozpoczęciem prac programistycznych określenie standardowego wyglądu i mechanizmów komunikowania się aplikacji oraz przetestowanie przyjętych założeń wraz z przyszłymi użytkownikami poprzez budowę prototypu.
- „mapa standardowych przypisań", która ustala w jaki sposób będą realizowane poszczególne elementy projektu np. za pomocą jakich środków będą realizowane funkcje bazy danych (procedury wbudowane, trigery, definicje więzów integralności, funkcje po stronie klienta).

Przedstawione powyżej standardy mogą stanowić *know-how* dostawcy i nie muszą być przedmiotem odbioru przez zamawiającego, natomiast zamawiający powinien się upewnić, że tego typu dokumentacja powstała i co więcej jest wykorzystywana.

Ze standardami ściśle związane są stosowane w przedsięwzięciu metodyki. Dostawca powinien ze swej strony zadbać o to aby zamawiający był w stanie obierać produkty związane z tą metodyką (plan projektu, specyfikacja wymagań, projekt rozwiązania docelowego), jak również o to aby zespół wykonawcy działał w sposób jednolity i zgodny z metodyką. Zwykle ten problem rozwiązywany jest cyklem szkoleń metodycznych przygotowujący zarówno zespół projektowy jak i użytkowników do stosowania przyjętej metodyki prac. Na tę kwestię należy zwrócić baczną uwagę w przypadku niejednorodnego zespołu projektowego (tzn. członkowie zespołu pochodzą z różnych organizacji i nie współpracowali wcześniej ze sobą). Należy uznać za groźną sytuację w której każdy z uczestników przedsięwzięcia stosuje swoją własną metodykę.

Ze stosowaniem metodyk bywa związany jeszcze jeden problem. Dostawcy często wskazują że w przedsięwzięciu będzie stosowana pewna „książkowa" metodyka, bez wnikania jak poszczególne elementy metodyki stosują się do konkretnego przedsięwzięcia. W praktyce każde przedsięwzięcie wymaga pewnej adaptacji metodyki i elementem planu projektu powinien być opis jak metodyka została zaadoptowana do danego przedsięwzięcia.

Literatura

Przedstawione poniżej pozycje mogą stanowić lekturę uzupełniającą do niniejszego opracowania, pozwalając na pełniejsze zrozumienie kwestii związanych z prowadzeniem przedsięwzięć informatycznych. Niestety nie ma na rynku pozycji w języku polskim odnoszących się bezpośrednio do zagadnień kontrolowania przedsięwzięć informatycznych, natomiast w podanych poniżej pozycjach można znaleźć elementy związane z kontrolowaniem i zapewnieniem jakości oraz wyrobić sobie na ich podstawie pogląd jak można (i powinno się) prowadzić przedsięwzięcia informatyczne.

[1] Gilb, Tom, Graham, Dorothy, *Software Inspection*, Addison—Wesley, 1993

[2] Grady, R.B., *Practical Software Metrics for Project Management and Process Improvement*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[3] Jones, Capers, *Patterns of Software Systems Failure and Success*, International Thomson Computer Press, 1996

[4] Weinnberg, Gerard M., Freedman Daniel P., *Walkthroughs, Inspections and Technical Reviews Evaluating Programs, Projects and Products*, Dorset House Publishing, 1990

Jan Zabrodzki

Politechnika Warszawska,
Instytut Informatyki

Grafika komputerowa

W czasie wykładu zostaną przedstawione informacje charakteryzujące obecny stan rozwoju grafiki komputerowej i tendencje jej rozwoju. Przedstawione zostaną slajdy reprezentatywne dla wybranych zastosowań grafiki komputerowej. W szczególności zostaną omówione takie zastosowania jak wizualizacja danych naukowych, sztuczna rzeczywistość, gry, symulatory, modelowanie zjawisk naturalnych, animacja w produkcji filmów, CAD itd.

Bogusław Jackowski

BOP s.c., Gdańsk

B.Jackowski@gust.org.pl

Znaki na niebie, ziemi i gdzie indziej

Dość długo obróbka żelaza pozostawała wierna stylowi i modelom odziedziczonym po epoce brązu, podobnie jak epoka brązu podtrzymywała początkowo kształty i stylistykę kamienia łupanego.

Mircea Eliade, *Kowale i alchemicy*

Rzut oka w przeszłość

Co zainspirowało człowieka przed tysiącami lat do pozostawiania znaków? Czy kształt stopy odciśnięty w miękkiej glebie? Czy ślad dłoni, przybrudzonej barwnikiem, pozostawiony na skale? Czy rysy, które pojawiały się w sposób naturalny na tworzywie pierwszych narzędzi? Trudno powiedzieć. Jednakże moment, w którym człowiek zaczął pozostawiać znaki celowo, skłonny byłbym uznać za narodziny cywilizacji. Znak – rozumiany jako nośnik przekazu, czyli symbol – jest nieodłącznym i niezwykle ważnym atrybutem kultury. To wynalazek znaku utorował drogę wynalazkowi pisma, bez którego trudno sobie wyobrazić rozwój cywilizacyjny.

Trwająca od kilku wieków eksplozja technologiczna wywarła istotny wpływ na rolę znaku w kulturze. Wprowadzenie maszyn zdolnych precyzyjnie powtarzać tę samą czynność zaowocowało tym, że znak zatracił swoją pierwotną indywidualność, zaczął ulegać – jakbyśmy to dziś powiedzieli – standaryzacji. Równocześnie człowiek – twórca idei znaku – odsuwany był przez postęp technologiczny coraz dalej od fazy nadawania znakowi ostatecznej postaci.

Twórcy paleolityczni najpewniej uczestniczyli w całym procesie tworzenia dzieła – od wyszukania i przygotowywania barwników aż do naniesienia ich własnoręcznie na skałę. Ten sposób tworzenia przetrwał bardzo długo, przez tysiąclecia proces przygotowywania narzędzi i tworzywa był udziałem twórcy. Tak było jeszcze w początkowej fazie powszechnej industrializacji. Na przykład artysta projektujący czcionkę sam wycinał jej kształt w metalu, sam składał tekst, sam parał się drukiem.

Później pojawiła się niezbędna w świecie podporządkowanym technologii specjalizacja, która spowodowała, że twórca nie miał szans, by brać udział w całym procesie powstawania znaków. Produkcją farb i pędzli zajęły się fabryki, projektant pisma przestał być drukarzem. Człowiek został oddzielony od swojego znaku.

Znak w dobie komputerów

Mniej więcej w połowie XX wieku wkroczył do akcji rzekomy sprzymierzeniec, a w istocie, jak się miało okazać, wróg znaku w jego dotychczasowej formie – komputer.

Otóż w komputerze znak nie jest po prostu znakiem. Ma swoją nader skomplikowaną strukturę, podobnie zresztą jak znak narysowany na kartce papieru, tylko że w przypadku znaku komputerowego struktury nie można w żaden sposób pominąć – i twórca, i użytkownik znaku o tej strukturze musi wiedzieć i umieć z tej wiedzy wyciągać wnioski. W przypadku znaku narysowanego na kartce papieru wystarczy wiedzieć, że papieru nie powinno się drzeć, wkładać do pieca, ani moczyć w wodzie. W przypadku znaku komputerowego trzeba wiedzieć znacznie więcej.

Najpospolitszym przykładem tego rodzaju struktury jest ponumerowanie liter alfabetu. Skądinąd ważne osiągnięcie intelektualne i technologicznie, dziś okazuje się hamulcem rozwoju i niewyczerpanym źródłem kłopotów. W ten sposób powstała bowiem dwoistość reprezentacji danych: dane mogą być zapisane bądź w postaci graficznej, gdzie rejestrowany jest jedynie kształt znaku, bądź w postaci „ponumerowanej", zwanej tekstową lub znakową.

Dwoistość taka jest absolutnie niezrozumiała dla tych, którzy z komputera korzystają sporadycznie i nie są zainteresowani zgłębianiem tajników informatyki. No jakże – argumentuje taki delikwent – przecież mam tutaj wydrukowany tekst, przygotowany zresztą na komputerze, pan ma skaner, to wrzuci pan ten tekst do komputera i już. My, szamani, wiemy, że potrzebny jest system OCR i śmieszy nas naiwność dyletanta. A nie powinna. Jego zdziwienie, gdy się dowiaduje, że gdyby nie wydrukował tego tekstu, tylko przyniósł dyskietkę, to sprawa przedstawiałaby się zupełnie inaczej, powinno nam dać do myślenia.

Dodatkowo gmatwa sprawę podział postaci graficznej na obwiedniową (ciągłą), tzn. zapisaną w postaci matematycznie przedstawionych krzywych, i rastrową (dyskretną), gdzie obraz jest pamiętany jako tablica pikseli[1] i każdemu pikselowi przypisany jest kolor. Pamiętajmy też, że grafika rastrowa daje się dalej podzielić na czarno-białą, szarą, paletową, RGB, CMYK i inne, a jakby tego było mało, to w odwodzie pozostają różne formaty graficzne (TIFF, GIF, JPEG) i różne sposoby kompresji danych (RLE, LZW, DCT).

Tu już dyletant nie ma szans się połapać, o co chodzi. Dlaczego tak samo wyglądający obrazek może zająć kilka kilobajtów lub kilka megabajtów? Dlaczego jeden obrazek daje się dowolnie skalować, a na drugim po przeskalowaniu pojawiają się jakieś wzorki? Szamani, mówiąc o tych wzorkach, używają określenia „mora", co wiele nie wyjaśnia – pewnie jak to szamani, lubują się w egzotycznie brzmiących słowach. Dlaczego wydrukowany obrazek jest nieostry, ba, nawet wyraźnie widać na nim jakieś kwadraciki, poszarpane brzegi, a na ekranie wszystko było dobrze? Szamani mówią coś o pikselach, niewłaściwej rozdzielczości, antyaliasingu – niech im będzie. Ale właściwie dlaczego nie można po prostu mieć sobie *znaku* w komputerze? Dlaczego ze znakiem w komputerze nie jest tak samo, jak ze znakiem narysowanym na kartce?

Na te pytania dyletant nie umie znaleźć odpowiedzi. I to też powinno dać nam do myślenia – przecież ma on prawo oczekiwać, że komputer ułatwi tworzenie i używanie znaku, a nie utrudni.

Zestawy znaków czyli fonty

Już pobieżny rzut oka na problem struktury pojedynczego znaku w komputerze może przyprawić o zawrót głowy. Łatwo sobie wyobrazić, co może oznaczać „struktura" w przypadku zestawów znaków, zwanych fontami. Wszystkie kłopoty związane z reprezentacją pojedynczych znaków przenoszą się na fonty, w szczególności zamieszanie związane z podziałem na odchodzące w przeszłość fonty rastrowe i przejmujące panowanie fonty obwiedniowe. Do tego dochodzą kłopoty specyficzne dla fontów.

Font to odpowiednik drukarskiej kaszty – zbiór znaków do drukowania tekstów. Dodatkowo font zawiera informacje metryczne o znakach – o ich rozmiarach, o pozycjonowaniu par znaków, o ligaturach itp. Informacje te zależą oczywiście od rodzaju (formatu) fontu. Obecnie najpopularniejsze formaty to windowsowy TrueType oraz postscriptowy Type 1; zapowiadane szumnie od jakiegoś

[1] Słowo *piksel* jest akronimem utworzonym ze słów *picture element*.

czasu wpólne dziecię firm Adobe[2] i Microsoft, mianowicie format OpenType, to – w uproszczeniu – „skrzyżowanie" tych dwóch formatów, wzbogacone o mechanizm zmiany układu znaków oraz parę gwizdków i dzwonków, związanych m.in. z ochroną praw autorskich.

Fonty odziedziczyły – zgodnie z powtarzającym się od pradziejów schematem – wiele cech po swoich przodkach, czyli kasztach drukarskich, w szczególności zestaw znaków, który jest znacznie bogatszy, niż zestaw oferowany przez maszyny do pisania, po których komputery odziedziczyły z kolei klawiatury. Takie niejednorodne dziedzictwo prowadzi nieuchronnie do zamieszania – nie może być inaczej, skoro drukarze dopracowali się np. czterech kresek poziomych różnej długości[3], natomiast na klawiaturze maszyny do pisania znalazła się tylko jedna kreska. Stąd w przytłaczającej większości tekstów przygotowywanych za pomocą komputerów pojawia się – wbrew możliwościom komputerów i tradycji drukarskiej – tylko najkrótsza z kolekcji kresek.

Dziedzictwo drukarskie w połączeniu z początkowo prymitywnymi implementacjami fontów zrodziło oprócz zamieszania także problemy natury estetycznej. Otóż znak we współczesnych fontach jest niejako „zamrożony" – jego kształt zasadniczo nie podlega modyfikacji. Niemniej jednak łatwość realizacji w komputerze przekształceń afinicznych spowodowała, że takie zabiegi jak skalowanie, zwężanie czy pochylanie znaków, weszły na stałe do repertuaru opcji programów do przetwarzania tekstów (tzw. edytorów formatujących). Nie trzeba chyba dodawać, że mistrzowie typografii nie chcą słyszeć o poddawaniu znaków transformacjom afinicznym. Według kanonów sztuki typograficznej, nawet zmianie rozmiaru znaku powinna towarzyszyć delikatna zmiana proporcji elementów znaku.

Tymczasem popularność, jaką cieszą się edytory formatujące, pchnęła ich twórców nawet dalej – większość edytorów potrafi wyliczyć z fontu podstawowego odmianę pogrubioną (w razie jej braku). Skutki estetyczne są opłakane, ale dzięki temu użytkownik nie musi się zastanawiać, dlaczego w danej rodzinie brakuje odmiany pogrubionej. Troskliwe zapobieganie powstawaniu refleksji w głowie użytkownika to tendencja ostatnich lat – celuje w tym oprogramowanie adresowane do tzw. masowego odbiorcy.

Niektórzy producenci fontów starali się jakoś przeciwdziałać złemu obyczajowi dokonywania prymitywnych zabiegów geometrycznych na znakach. Godną uwagi próbą jest opracowanie przez firmę Adobe formatu fontów postscriptowych Multiple Master. Font typu Multiple Master zawiera w istocie kilka osobno zaprojektowanych odmian danej rodziny, np. normalną, wąską i szeroką. Użytkownik systemu potrafiącego obsłużyć fonty Multiple Master może zażyczyć sobie, by jego dokument został złożony fontem będącym dowolną kombinacją liniową fontów[4] stanowiących bazę fontu Multiple Master. Dzięki temu można uzyskać font lekko poszerzony lub zwężony z zachowaniem stosownych zmian proporcji elementów znaku. Jest to niewątpliwie słuszny krok w stronę „odmrożenia" znaku w foncie – wraz z rosnącą mocą komputerów znaki mogą być opisywane w coraz bardziej złożony sposób. Fonty Multiple Master są jednak skomplikowane w produkcji, dość niewygodne w obsłudze i niestety ograniczone jeśli chodzi o możliwości – stąd zapewne ich umiarkowana popularność.

Inne próby „ożywienia" znaków w foncie, takie jak na przykład delikatna zmiana kształtu szeryfów przy niektórych zestawieniach liter, nie wyszły – z tego co wiem – poza laboratoria typograficzne. (Cudactwo, zaimplementowane w najświeższych wersjach edytora Word, mianowicie fonty animowane, nie ma oczywiście nic wspólnego z „ożywianiem" znaków, o którym tu mowa.)

Wspomniane już fonty OpenType nie stanowią w tej mierze przełomu, mimo nieustannego podkreślania przez firmy

[2] Dziełem firmy Adobe jest język PostScript, będący światowym standardem opisu strony graficznej.

[3] Są to: dywiz – najkrótszy, pauza półfiretowa i pozycjonowany na innej wysokości matematyczny minus – oba znaki średniej, acz z reguły nieco różnej długości, oraz pauza firetowa – najdłuższa.

[4] Pojęcie kombinacji liniowej daje się zastosować do krzywych Béziera, z których zbudowane są znaki w fontach postscriptowych, daje się więc przenieść na znaki, a tym samym także na fonty.

zaangażowane w przedsięwzięcie, że OpenType może bez mała wszystko.

Standardy kodowania czyli wieża Babel

Ale zamieszanie wywołane skomplikowanym rodowodem komputera to pół biedy. Aspekty estetyczne też zawsze można zrelatywizować. Gorzej, że na scenie pojawia się koń trojański numerowania znaków. Przez długie lata liczba znaków w foncie na mocy tradycji, podpieranej argumentem rzekomej efektywności, nie przekraczała magicznej wartości 256. Póki technologia komputerowa rozwijała się w krajach, które nie używają znaków diakrytycznych, problem pozostawał w ukryciu. Niestety szybko okazało się, że inne kraje też chcą używać komputerów i muszą w związku z tym przyporządkować numery swoim znakom. Nagle w 256-znakowej tablicy zrobiło się za ciasno dla wszystkich. Jak grzyby po deszczu zaczęły więc pojawiać się „standardy" kodowania, uwzględniające to te, to tamte znaki i – jak pokazała praktyka – nigdy nie zaspokajające w pełni potrzeb użytkowników.

Istnienie setek schematów kodowania nasuwa nieodparcie analogię z wieżą Babel, aczkolwiek w tym przypadku „pomieszanie języków" okazało się (na razie) mniej brzemienne w skutki. Mimo powszechnego zastosowania komputerów do komunikacji można się jakoś dogadać, ot, najwyżej polskie znaki zostaną pozbawione elementów diakrytycznych albo zamienione na dziwne symbole, zwane pieszczotliwie krzaczkami. *Polskie znaki można uzyskać tylko pod Windowsem, a wysyłać spod Windowsa nie potrafię* – oznajmił mi w liście wysłanym pocztą elektroniczną pewien młody adept sztuki komputerowej. Pióro i papier nie przysparzały aż takich kłopotów, prawda?

Ale babelopodobne zamieszanie nie niepokoiłoby mnie, gdyby nie trendy, wskazujące na umacnianie się wadliwej od początku koncepcji numerowania znaków w sposób uniwersalny. Skoro jeden bajt nie wystarcza do ponumerowania, to użyjmy dwóch bajtów – jakie to proste! Za proste, żeby mogło zadziałać.

Koncepcja ta uzyskała rangę normy ISO/IEC 10646 i znana jest pod nazwą standardu Unicode. Nie podważając celowości tego rodzaju przedsięwzięć standaryzacyjnych, odnoszę się z rezerwą do upatrywania w standardzie Unicode panaceum na bolączki związane z komputerowym przetwarzaniem tekstów.

Po pierwsze, standard Unicode raczej ukierunkowany jest na znaki (*characters*) a nie na formy (*glyphs*). W związku z tym takie twory jak ligatury, jako nie mające tożsamości semantycznej, nie należą do obszaru zainteresowań Unicode. Ligatury tradycyjne (takie jak „fi", „ff", „fl", „ffi" czy „ffl") znalazły się wprawdzie w tabelach Unicode, ale dziś znawcy tematu coraz bardziej skłaniają się ku uznaniu tej decyzji za błędną. Tymczasem typografia, zwłaszcza wytworna, nierozerwalnie wiąże się z wykorzystaniem różnorakich, niekoniecznie standardowych ligatur. Zatem brak miejsca na formy tego rodzaju w standardzie Unicode oznacza, że jest to w najlepszym razie adekwatny standard językowy, ale nie typograficzny. A zważmy, że dzięki gwałtownemu rozwojowi systemów DTP zastosowania typograficzne są – i wolno przypuszczać, że pozostaną – niezwykle popularne.

Po drugie, Unicode w sposób zdecydowany odcina się od języków „nietekstowych", takich jak notacja muzyczna czy taneczna. Jeśli więc Unicode zapanuje jako standard wymiany dokumentów, ważne obszary zastosowań, w których wykorzystywane są znaki „nietekstowe", w sposób sztuczny zostaną wyniesione poza standard. Utrudni to życie tym, którzy mają nieszczęście zajmować się notacją muzyczną czy taneczną. Notacja muzyczna i taneczna, a także pismo kaligraficzne to niezwykle ciekawe przykłady języków posługujących się w istocie *nieskończoną* liczbą znaków. Nie dziwota, że fonty zbudowane z „zamrożonych" znaków i różne systemy numerowania znaków nie pasują do tych języków. Nie oznacza to jednak, że komputer jako taki jest niedobrym narzędziem do składu nut czy zapisu kroków tanecznych.

Po trzecie, języki takie jak tolkienowski Tengwar czy język klingoński z serialu StarTrek z trudem mieszczą się w zakresie zainteresowań standardu Unicode. Oba wymienione języki być może znajdą się w standardzie Unicode, co z jednej strony wydaje się niezupełnie poważne, a z drugiej strony długotrwała z natury procedura

włączania zaproponowanego języka do standardu Unicode powoduje, że twórcy nowych języków, np. fantastycznych, mogą się pożegnać z myślą o wygodnym posługiwaniu się standardem Unicode.

Po czwarte wreszcie, sporo fontów nie zawiera liter, tylko znaki ozdobne lub ornamenty. Takie znaki powinny znaleźć się w obszarze przeznaczonym na użytek prywatny – zarezerwowane są na to kody od U+E000 do U+F8FF (6400 znaków), gdzie można umieszczać swoje własne znaki pod własnymi nazwami i kodami. Jest to ewidentna próba załatania problemu, i to niezbyt skuteczna, bo istniejące narzędzia (Windows NT, Windows 98) nie umożliwiają wprowadzania takich znaków z klawiatury, nie ma bowiem mechanizmu swobodnego definiowania klawiszy, czy też kojarzenia klawiatury z fontem. W związku z tym należy przewidywać, że jeszcze długo znakom ozdobnym będą przypisywane kody standardowo dostępne z klawiatury, przede wszystkim kody liter i cyfr.

Jakim prawem?

Wraz z rozwojem oprogramowania komplikował się opis znaku. I jeśli w przypadku grafiki rastrowej można było mówić o znaku jako o danych do programu wyświetlającego lub drukującego znak, to obecnie nie jest jasne, czy znak to program, czy dane. W przypadku znaku opisanego za pomocą języka PostScript raczej trzeba by mówić o programie, chociaż z kolei przygotowywanie znaku za pomocą programu takiego jak np. CorelDRAW trudno uznać za programowanie. Ale skoro PostScript jest językiem programowania, to można sobie wyobrazić znaki opisywane bezdyskusyjnie za pomocą programów.

Jeszcze bardziej sytuacja się komplikuje w przypadku znaków animowanych, ostatnio niezwykle popularnych za sprawą Internetu. Czy jest to program, czy dane? Jeśli chodzi o animowane obrazki w formacie GIF, to jeszcze można by się skłaniać ku interpretowaniu serii obrazów jako zbioru danych, ale już w przypadku wirtualnej grafiki zapisanej w języku VRML można mieć zasadnicze wątpliwości.

Rodzi to problemy natury prawnej, związane z ochroną praw autorskich. Rozwój pojęć prawnych wyraźnie nie nadąża za rozwojem informatyki – świat software'u przestał się dzielić na programy i dane. Tymczasem prawo inaczej chroni programy, a inaczej pozostałe utwory autorskie. Konieczne jest zatem zaklasyfikowanie, czy dany utwór (w naszym przypadku znak) jest programem czy też nie. Prawnicy twierdzą bowiem, że jest niedopuszczalne, by ten utwór sam w jednym przypadku był chroniony w sposób specjalny jak program komputerowy, a w innym – na zasadach ogólnych.

Jeszcze trudniej jest z określeniem, czym jest z prawnego punktu widzenia współczesny font obwiedniowy (na przykład TrueType). Dane? Nie. Program? Nie. Biblioteka programów? Nie. Baza danych? Też raczej nie, choć gdybym musiał się w tej kwestii wypowiedzieć, to określiłbym font jako coś w rodzaju bazy danych, która zawiera programy opisujące kształty poszczególnych znaków.

Mimo niejasnej merytorycznie sytuacji prawnicy skłaniają się ku uznaniu fontu za utwór podlegający takiej ochronie jak program, niemniej jednak co jakiś czas podnoszone są wątpliwości, czy analogia między fontem a programem jest istotnie pełna.

Wątpliwości dochodzą do głosu szczególnie wyraźnie w przypadku kwestii praw autorskich do znaków dodanych lub zmodyfikowanych. Typowym zabiegiem jest dodanie do fontu znaków narodowych[5] i specjalnych. Powracająca jak bumerang na łamy prasy dyskusja o „prawnej ochronie ogonków" świadczy o niejasnym statusie prawnym takich zabiegów.

Bardzo poważnym zagadnieniem, zresztą nie tylko z prawnego punktu widzenia, jest dołączanie grafiki lub fontów do dokumentu.

[5] Zmiana fontu bez zgody właściciela praw autorskich jest oczywiście nielegalna. Jednak często bywa tak, że zakupione oprogramowanie jest wprawdzie przystosowane do języka danego kraju, natomiast dystrybuowane wraz z nim fonty – nie. Stawia to użytkownika w sytuacji prawnie dwuznacznej, nie bardzo wiadomo bowiem, kto i za co ponosi odpowiedzialność: czy użytkownik za ewentualne zmodyfikowanie fontu, czy producent za sprzedaż wadliwego towaru, czy też – jak zdaje się wskazywać praktyka – nikt za nic nie odpowiada.

Obecnie w większości przypadków odbiorca dokumentu jest w stanie wyłuskać z przesłanego materiału zarówno font, jak i grafikę. W przypadku grafiki sytuacja prawna na ogół jest dość jasna, gdyż grafika jest chroniona na zasadach ogólnych. Font natomiast, jako „utwór komputerowy", podlega specjalnym obwarowaniom prawnym i jeżeli odbiorca dokumentu nie nabył fontu, to nie ma prawa do jego posiadania, co uniemożliwia praktycznie legalne przekazanie dokumentu z dołączonym fontem. Z kolei niedołączenie fontu bardzo często prowadzi do katastrofy, zwłaszcza w przypadku dokumentów sformatowanych (PDF), aczkolwiek dokumenty, w których formatowanie odgrywa drugorzędną rolę (HTML), również mogą płatać figle na skutek użycia różnych fontów przez odbiorcę i nadawcę dokumentu.

Niektóre aplikacje tworzące dokumenty PDF dołączają fonty w postaci fragmentarycznej (*partial downloading*), tzn. dołączane są jedynie znaki wykorzystywane w dokumencie. W większości przypadków font taki po wyłuskaniu jest praktycznie nieużywalny, co stanowi niejaką ochronę. Wspomniany format OpenType określa, czy font może zostać dołączony do dokumentu, czy może zostać dołączona jedynie fragmentaryczna informacja pozwalająca na wydruk i podgląd, czy też dokument może zawierać kompletny font w wersji instalowalnej. Jest to krok w stronę ochrony prawnej fontów – czy wystarczający, pokaże przyszłość.

Podejrzewam jednak, że wraz z rozwojem informatyki jej sytuacja prawna będzie ulegała – przynajmniej w najbliższej przyszłości – pogarszaniu, to znaczy przeciwdziałanie przestępstwom będzie coraz mniej skuteczne. Cała nadzieja w tym, że złoczyńców nigdy nie jest zbyt wielu.

Rzut oka w przyszłość

Jaki los zgotuje znakom nadchodzące tysiąclecie? Bacznie obserwując znaki na niebie można dostrzec dwa scenariusze.

W pierwszym scenariuszu znaki byłyby całkowicie odseparowane od człowieka, tzn. człowiek byłby biernym użytkownikiem gigantycznej liczby *clipartów*, zrobionych przez nie wiadomo kogo, nie wiadomo na jaką okazję, zapisanych w ściśle tajnym formacie. Styczność użytkownika ze znakiem sprowadzałaby się do naciśnięcia klawisza myszy. Jeśli chodzi o fonty, to ten wariant jest już właściwie realizowany – użytkownik dostaje font w postaci czarnej skrzynki, niekonfigurowalnej, dającej się używać w jedynie słuszny sposób przewidziany przez producenta fontu.

Zresztą producent na skutek starannego ukrywania informacji przed użytkownikiem zaczyna w końcu sam przed sobą tę informację chować, tracąc panowanie nad swym dziełem. Dowodem na taki stan rzeczy jest „specyfikacja" formatu TrueType opublikowana przez Microsoft – jest to najgorsza, najbardziej mętna dokumentacja, jaką zdarzyło mi się trzymać w rękach; w grę wchodzi jedynie wzięcie dokumentacji do ręki – o czytaniu mowy nie ma. Takiej polityce dokumentacyjnej przypisałbym ciągnące się po dziś dzień zapluskwienie oprogramowania przetwarzającego fonty TrueType – niemalże wszystkie znane mi tzw. profesjonalne aplikacje fontowe generują wadliwe pliki TTF, skutkiem czego użytkownicy fontów TrueType mogą liczyć na przeróżne niespodzianki, nie zawsze miłe.

Według drugiego scenariusza człowiek znalazłby się ponownie nieco bliżej znaku. Specjalizacji przeskoczyć się nie da, ale można sobie wyobrazić, że świadomy użytkownik potrafiłby dokonać stosownej rekonfiguracji znaku czy też fontu (legalnie, zgodnie z zamierzeniami twórcy), aktywnie uczestnicząc w procesie tworzenia. Patrząc optymistycznie można oczekiwać, że wykluje się kiedyś standard opisu „inteligentnych znaków" – nie byłyby one już zamrożoną formą, ale zmieniałyby na życzenie użytkownika swój kształt w ściśle określonym zakresie w zależności od potrzeb. Zbudowany z takich znaków „inteligentny font" byłby czymś w rodzaju małego systemu składu, kontaktującego się w sposób konfigurowalny z klawiaturą i z systemem, nie mniej inteligentnym od fontu. Już widzę, jak dziadkowie opowiadają wnuczkom, że kiedyś, w epoce komputera „kamiennego" próbowano brutalnie znaki w ponumerowanych klatkach prostokątnych zamykać.

Żeby nie zabrnąć za daleko w science-fiction, spójrzmy na ten świetlany obraz przez pryzmat tradycji czcionki z ołowiu, ubogiej

klawiatury maszyny do pisania, numeracji znaków, skomplikowanej reprezentacji grafiki, utrwalającego się obyczaju tworzenia oprogramowania promującego bezmyślność użytkownika, itp.. itd. Tradycja, jak już zostało wspomniane, bywa w technologii niezwykle silnym hamulcem rozwoju. Nie należy więc oczekiwać w najbliższym czasie rewolucyjnych zmian. Może to i dobrze, byle tylko ewolucja nam się nie zatrzymała.

Tomasz Traczyk

Instytut Automatyki i Informatyki Stosowanej
Politechniki Warszawskiej
e-mail: ttraczyk@ia.pw.edu.pl
URL: http://www.ia.pw.edu.pl/~ttraczyk/

Bazy danych, narzędzia dostępu do danych i hurtownie danych

Fundament dużej części współczesnych systemów informatycznych stanowią bazy danych. Pokazano czym właściwie jest baza danych i dlaczego zastosowanie systemu zarządzania bazą danych jest właściwym rozwiązaniem bardzo wielu typowych problemów informatycznych. Omówiono podstawowe praktycznie ważne modele baz danych. Przedstawiono podejścia do tworzenia systemów z bazami danych i typowe narzędzia dostępu do danych. Dokładniej opisano stosunkowo nową, ale ważną kategorię systemów z bazami danych: hurtownie danych.

1 Bazy danych

W ogromnej większości systemów informatycznych zasadnicze znaczenie ma bezpieczne przechowywanie i sprawne przetwarzanie danych. Jeśli danych jest dużo i musi z nich korzystać wielu użytkowników, na ogół właściwym rozwiązaniem jest zastosowanie bazy danych. Ponieważ takie wymagania są typowe, bazy danych są stosowane powszechnie.

1.1 Co to jest baza danych?

Bazą danych nazywa się zwykle system danych, który przechowuje dane w pamięci trwałej (np. na nośniku magnetycznym), ma zdefiniowaną strukturę i jest wyposażony w tzw. reguły integralności, określające warunki poprawności danych.

By taki system danych miał znaczenie praktyczne, musi posiadać liczne dodatkowe cechy, które omówiono w następnym punkcie. Zapewnić je można budując nad systemem danych odpowiednie oprogramowanie, które będzie obsługiwało dane w sposób zapewniający potrzebne nam właściwości. Nazywa się ono systemem zarządzania bazą danych (DBMS — *Database Management System*). Podstawową cechą i rolą DBMS jest izolowanie programisty i tworzonych przez niego aplikacji od fizycznej reprezentacji danych (ich organizacji w pamięci, na dyskach itp.). Dane, dostępne przez odpowiednie interfejsy, postrzegane są przez programistę jako zorganizowane w sposób, który nazywamy modelem danych.

1.2 Znaczenie baz danych

Dlaczego bazy danych stanowią podstawową platformę budowy profesjonalnych systemów informatycznych? Odpowiedzi należy szukać z zestawie ważnych cech, które typowo zapewniają systemy zarządzania bazami danych

Owe cechy to:

- *Niezawodność zapisu* — dane zapisane w bazie nie mogą być ulotne, tzn. nieodporne na awarie. Z reguły zapewnione są mechanizmy wykonywania kopii rezerwowych, automatycznego odtwarzania danych po mniejszych awariach (np. zasilania) i odtwarzania z kopii rezerwowych po awariach groźniejszych (np. nośnika).
- *Integralność danych* — dane zapisane w bazie muszą być poprawne, tzn. spełniać zdefiniowane w bazie reguły integralności; dane nie spełniające tych reguł nie mogą być zapisane. Poprawność danych powinna być kontrolowana przez sam DBMS, a nie przez aplikacje.
- *Sprawność zapytań* — DBMS musi zapewniać wydajne wyszukiwanie danych. Najczęściej odbywa się to przez tzw. język zapytań, w którym deklaruje się warunki, jakie mają spełniać wyszukiwane dane. Zapytanie winno być przez DBMS wykonane w możliwie najszybszy sposób, co wymaga tzw. optymalizacji zapytania.
- *Wygodne interfejsy* — dane muszą być dostępne dla wielu różnych programów, napisanych za pomocą różnych języków i narzędzi. DBMS powinien zapewnić dostęp do danych:
  - z programów napisanych w językach programowania ogólnego przeznaczenia (3GL),
  - z aplikacji napisanych w językach specjalizowanych i językach czwartej generacji (4GL),
  - z typowych programów biurowych (arkusze kalkulacyjne, edytory tekstu),
  - z WWW i języka Java.
- Wielodostęp — z reguły systemy informatyczne wykorzystujące bazy danych są przeznaczone do użycia przez wielu użytkowników jednocześnie. DBMS musi dać im wydajny dostęp do danych i zapewnić, by operacje wykonywane równocześnie przez różnych użytkowników nie przeszkadzały sobie wzajemnie.
- Zabezpieczenia dostępu — dane zgromadzone w bazach danych mają na ogół duże znaczenie i muszą być odpowiednio chronione. DBMS musi zapewnić mechanizmy autoryzacji dostępu do bazy danych i możliwość zróżnicowania uprawnień różnych użytkowników.

Cechy te są niezbędne w większości systemów informatycznych. Budowanie na fundamencie sprawdzonego systemu zarządzania bazą danych pozwala te cechy osiągnąć stosunkowo niedużym kosztem i z gwarancją niezawodności.

1.3 Współczesne modele baz danych

Dane udostępniane przez DBMS są zorganizowane (z punktu widzenia użytkownika) według pewnych ogólnych zasad, zwanych modelem danych. Zgodnie z tym modelem tworzone są szczegółowe struktury danych, tworzące tzw. schemat danych, odpowiadający konkretnemu systemowi informatycznemu. Schemat danych jest odwzorowany w tzw. słowniku danych, udostępnianym przez DBMS.

Współcześnie znaczenie praktyczne ma kilka modeli danych[1].

1.3.1 Model relacyjny

Ogromna większość współcześnie projektowanych i działających systemów funkcjonuje w oparciu o relacyjny model danych i systemy zarządzania relacyjnymi bazami danych (RDBMS — *Relational DBMS*).

W modelu relacyjnym dane są zapisywane w tabelach. Każda tabela jest nazwana i ma określoną, stałą liczbę kolumn. Kolumny są opatrzone nazwami i mogą przechowywać dane określonego prostego typu (np. liczby o podanej precyzji albo napisy o określonej długości). Liczba wierszy w tabeli jest zmienna, można je dodawać lub usuwać.

Jednostka danych — wiersz — nie ma żadnej stałej tożsamości i jest identyfikowana jedynie przez wartości danych zapisanych w wierszu. Zwykle dla identyfikacji wierszy

[1] Nie omówiono modeli: hierarchicznego i sieciowego, choć nadal wielkie ilości danych są przechowywane w takich systemach. Nowe systemy tego typu nie są już jednak tworzone.

definiuje się pewien zestaw kolumn, w którym zawarte są dane tworzące niepowtarzalne kombinacje — taki zestaw kolumn nazywa się kluczem głównym. Dane zapisane w poprawnie zbudowanej relacyjnej bazie danych z reguły są podzielone na wiele tabel. Związki między danymi zawartymi w różnych tabelach ustanowione są przez równość wartości wpisanych w pewnych wyróżnionych kolumnach: kluczu głównym w tabeli „nadrzędnej" i tzw. kluczu obcym w tabeli „podrzędnej".

Niepowtarzalność wartości klucza głównego oraz zgodność wartości klucza obcego z istniejącą wartością klucza głównego stanowią podstawowe reguły integralności (tzw. integralności referencyjnej), których spełnienie jest nadzorowane przez RDBMS.

Niekorzystnym zjawiskiem, mogącym wystąpić w bazie danych, jest redundancja, czyli powtarzanie danych. Redundancja powoduje marnowanie miejsca oraz — co znacznie ważniejsze — występowanie tzw. anomalii przy operacjach na danych (np. możliwości modyfikacji tylko jednego z wielu wystąpień powtórzonych danych).

W przypadku relacyjnych baz danych dąży się na ogół do budowania tzw. znormalizowanych schematów danych, spełniających pewne teoretycznie uzasadnione warunki, zapewniające eliminację redundancji i różnego rodzaju anomalii. Normalizacja schematu prowadzi z reguły do podziału danych na bardzo wiele tabel połączonych licznymi związkami. Jest to, niestety, reprezentacja mało naturalna i — w przypadku danych złożonych — kłopotliwa w przetwarzaniu.

Wielką zaletą relacyjnych baz danych jest jednak prostota podstawowych struktur danych — tabel. Umożliwiło to zbudowanie efektywnych i niezawodnych systemów zarządzania bazami danych i skonstruowanie stosunkowo prostych języków zapytań. Ważnym atutem relacyjnych baz danych jest istnienie powszechnie stosowanego standardowego języka zapytań SQL (*Structured Query Language*).

Praktyczne znaczenie relacyjnych baz danych jest ogromne, a ich pozycja nie wydaje się zagrożona. O wielkiej popularności rozwiązań relacyjnych decyduje dostępność niezawodnych i wydajnych RDBMS o ugruntowanej pozycji rynkowej. Ważnym czynnikiem jest także dobre dostosowanie struktur relacyjnych do reprezentowania danych ekonomicznych i finansowych, co ciągle stanowi istotę większości dużych systemów informatycznych.

1.3.2 Model obiektowy

Model relacyjny niezbyt dobrze nadaje się do przechowywania i przetwarzania danych złożonych, np. multimedialnych, hipertekstowych, opisujących projekty techniczne lub mapy itp. Reprezentacja relacyjna takich danych jest z reguły bardzo skomplikowana, a wydajność przetwarzania — mierna. Model relacyjny jest także stosunkowo mało podatny na modyfikacje. Zwłaszcza zmiany dotyczące kluczy głównych i obcych są kłopotliwe i prowadzą do konieczności bardzo znacznej przebudowy aplikacji.

Wad tych nie ma model obiektowy. W modelu tym dane podzielone są na tzw. obiekty, odpowiadające opisywanym obiektom w świecie rzeczywistym. Każdy obiekt należy do klasy, będącej zbiorem obiektów tego samego typu. Definicja klasy obejmuje definicję struktury danych obiektu (tzw. atrybutów) oraz operacji na tych danych.

Każda jednostka danych — obiekt — posiada niezmienną tożsamość niezależną od zawartości obiektu. Budowanie związków między obiektami jest zatem proste, a związki są odporne na modyfikacje zarówno wartości danych jak i struktury danych.

W obiektowym modelu danych zastosowano osiągnięcia programowania obiektowego i wprowadzono liczne istotne możliwości, z których najważniejsze omówiono poniżej.

- *Złożone typy danych*: Dane w modelu obiektowym mogą mieć złożone typy (krotki, kolekcje itp.), budowane z typów prostych i innych typów złożonych. Twórca schematu sam definiuje potrzebne typy danych.
- *Operacje definiowane wraz z danymi:* Razem danymi można zdefiniować kod operacji (tzw. metody) na tych danych.

- *Dziedziczenie:* Kolejne klasy można definiować używając już istniejących definicji i uzupełniając je o nowe atrybuty i operacje.
- *Enkapsulacja:* Dane i operacje zawarte w danym obiekcie można podzielić na dostępne z zewnątrz oraz dostępne tylko z metod tego obiektu. W ten sposób część lub całość struktury obiektu można osłonić przed bezpośrednim dostępem, pozostawiając możliwość oddziaływania na wnętrze obiektu za pomocą jego metod.
- Polimorfizm: Ta sama nazwa może być użyta do nazwania operacji w różnych klasach, w tym w klasach od siebie dziedziczących. Można zatem pisać procedury, używające tak nazwanych operacji, które działają poprawnie dla wielu różnych typów danych — rzeczywiste działanie programu zależy bowiem od klasy konkretnego obiektu, na którym procedura działa.

Specyficzne cechy obiektowych baz danych powodują, że są one bardzo dobrze przystosowane do obsługi danych złożonych i są w dużym stopniu podatne na modyfikacje.

Niestety, implementacja modelu obiektowego jest znacznie trudniejsza niż w przypadku modelu relacyjnego. Model ten jest także trudniejszy koncepcyjnie oraz mało znany projektantom i programistom. Nie ma też jednego, powszechnie uznanego za standard, modelu danych ani języka zapytań (choć istnieją godne uwagi propozycje).

Dostępne są liczne systemy zarządzania obiektowymi bazami danych (OODBMS — *Object Oriented DBMS*), ale żaden z nich nie zdobył dostatecznie silnej pozycji na rynku zastosowań komercyjnych. Obiektowe bazy danych są obecnie spotykane w zastosowaniach eksperymentalnych i niszowych.

1.3.3 Model relacyjno-obiektowy

W ostatnim czasie pojawiły się na rynku rozwiązania próbujące połączyć najważniejsze zalety modelu relacyjnego i obiektowego.

Trend ten wiąże się z faktem ogromnego rozpowszechnienia relacyjnych baz danych. Radykalna zmiana modelu danych w już istniejących systemach nie jest możliwa przy rozsądnych nakładach. Tymczasem model relacyjny ma ograniczenia, które bardzo utrudniają jego stosowanie w wielu rozwijających się dynamicznie dziedzinach.

Postawiono zatem na rozwiązania ewolucyjne: relacyjna struktura danych powinna być rozbudowana o najważniejsze elementy modelu obiektowego.

W modelu relacyjno-obiektowym dane są zorganizowane w tabele, ale w kolumnach oprócz danych prostych typów mogą się także znaleźć obiekty złożone. Typy obiektów złożonych powinny dać się definiować. Pożądana jest także możliwość definiowania operacji (metod) związanych z obiektami. Identyfikację danych, istniejącą w bazach relacyjnych (klucze główne), wzbogaca się o mechanizmy niezmiennej tożsamości obiektów i możliwość ustanawiania związków opartych na takiej tożsamości.

Model relacyjno-obiektowy jest stosunkowo nowy, a systemy zarządzania takimi bazami danych (ORDBMS — *Object-Relational DBMS*) są jeszcze nieliczne i mają wiele ograniczeń, np. brak dziedziczenia i hermetyzacji, konieczność programowania metod poza bazą danych w zewnętrznym języku programowania itd.

Bazy relacyjno-obiektowe mogą zaspokoić wiele oczekiwań użytkowników, związanych z nowymi wyzwaniami stojącymi przed systemami informatycznymi. Choć nie mają wszystkich zalet modelu obiektowego, dają za to możliwość ewolucyjnego rozwoju już istniejących systemów, przy zachowaniu dotychczasowych inwestycji. W przygotowaniu jest nowy standard języka zapytań dla takich baz (SQL3), będący rozszerzeniem powszechnie znanego języka SQL. Wydaje się, że ORDBMS wyznaczają trend rozwojowy, który ma największe szanse na rynku.

2. Narzędzia dostępu do baz danych

Dane zgromadzone w bazie danych i zarządzane przez DBMS muszą być udostępnione użytkownikom przez różnego rodzaju interfejsy i aplikacje. DBMS ze schematem danych oraz zestaw aplikacji i interfejsów

tworzą razem „bazodanowy" system informatyczny.

2.1 Składniki systemu z bazą danych

Typowy system z bazą danych zawiera wiele różnych składników: DBMS ze schematem danych, warstwę sieciową (zależnie od architektury systemu), aplikację realizującą główne zadania systemu, procedury przekazywania danych w postaci elektronicznej z/do systemu, narzędzia pomocnicze: do administrowania, zapytań *ad hoc* itd.

Aplikacja, która realizuje główne zadania systemu, zwykle składa się z menu sterującego, formularzy (ekranów) służących do interaktywnych operacji na danych i raportów służących do sporządzania wydruków. Aplikacja może zawierać także inne elementy, np. wywołania procedur przetwarzających dane.

Każdy ze składników takiego systemu informatycznego może być wytworzony za pomocą odpowiedniego specjalizowanego oprogramowania narzędziowego.

2.2 Kategorie systemów z bazami danych

Spotykane na rynku systemy zarządzania bazami danych[2] można podzielić na kilka kategorii „wagowych":

- Systemy biurkowe (*desktop*) — są przeznaczone do pracy indywidualnej na niewielkich zbiorach danych lub do bardzo małych biur, nie są zwykle wyposażone we wszystkie cechy profesjonalnego DBMS.
- Systemy osobiste (*personal*) — mają zwykłe cechy „dojrzałych" DBMS, ale są przeznaczone do pracy jednostanowiskowej, np. dla projektantów systemów albo osób pracujących na komputerach przenośnych bez dostępu do sieci.
- Systemy dla grup roboczych (*workgroup*) — są przeznaczone dla ograniczonej liczby równocześnie pracujących użytkowników; poza tym mają większość cech pełnowartościowych DBMS.
- Systemy dla przedsiębiorstw (*enterprise*) — wyposażone są we wszelkie mechanizmy umożliwiające pracę przy masowym wielodostępie (tysiące użytkowników), na wielkich zbiorach danych (terabajty), w sieci lokalnej i rozległej.

Oferta największych producentów baz danych obejmuje zwykle DBMS z większości tych grup, na ogół zgodne ze sobą w stopniu umożliwiającym łatwe przeniesienie (skalowanie) istniejącego systemu z jednej kategorii „wagowej" do innej.

2.3. Architektury systemów z bazami danych

Systemy z bazami danych budowane są w różnych architekturach, zależnych od potrzeb i zastosowanego sprzętu.

2.3.1 Architektury małych systemów

Dla systemów biurkowych i osobistych typowa jest architektura jednostanowiskowa, gdy baza danych jest dostępna tylko dla jednego użytkownika, a DBMS i aplikacja dostępu do danych działają na tym samym komputerze.

Niewielkie systemy do małych biur buduje się także w architekturze sieciowej-plikowej, gdzie kilku użytkowników korzysta ze wspólnych danych przez wielodostęp do plików. Każdy z użytkowników na swoim komputerze uruchamia osobną kopię DBMS i aplikacji, a dane znajdują się w jednym miejscu, w plikach dostępnych przez sieć lokalną.

2.3.2 Architektura terminalowa

Architektura terminalowa jest typowa dla systemów zbudowanych w oparciu o komputery klasy mainframe i minikomputery oraz wczesne systemy uniksowe. W takich systemach DBMS oraz aplikacje działają na centralnym komputerze, a użytkownicy mają do dyspozycji nieinteligentne terminale

[2] Dalsze rozważania odnoszą się do relacyjnych baz danych.

(najczęściej znakowe). Architekturę tę spotyka się głównie w dużych instytucjach, np. w bankach.

2.3.3 Architektura klient-serwer

Obecnie najpowszechniej spotykana jest architektura klient-serwer. DBMS działa tu na centralnym komputerze — serwerze, zaś aplikacje działają na komputerach klientów. Rolę serwerów pełnią najczęściej systemy uniksowe, zaś stacjami klientów są komputery PC albo stacje robocze.

Na serwerze odbywa się przechowywanie danych, wykonywanie zapytań i przetwarzanie danych, natomiast stacje klientów wykonują logikę aplikacji (w tym obsługę złożonego interfejsu graficznego typu GUI) i niekiedy także elementy przetwarzania danych.

Architektura klient-serwer jest dobrze dostosowana do możliwości współczesnych małych (i niedrogich) komputerów, daje duże możliwości rozbudowy i skalowania. Podłączenie nowego klienta jest tu bardzo łatwe i zwykle nie wymaga specjalnego inwestowania w sprzęt, bowiem nowy użytkownik na ogół już ma komputer PC. Potrzebna jest natomiast z reguły instalacja odpowiedniego oprogramowania.

Architektura ta świetnie przystaje do sposobu przetwarzania właściwego dla SQL-owych relacyjnych baz danych. W tym modelu przetwarzania zapytania (krótkie teksty) są bowiem wysyłane do serwera danych, a serwer odpowiada odsyłając jedynie te (na ogół nieliczne) dane, które są potrzebne; nie jest więc konieczne masowe przesyłanie danych przez sieć.

Systemy klient-serwer oszczędnie wykorzystują sieć, a komputer centralny jest odciążony od zadań, które z powodzeniem mogą być wykonane na klientach — typowych komputerach PC.

2.3.4 Architektura trójwarstwowa

Podstawową wadę architektury klient-serwer stanowią znaczne koszty użytkowania komputerów klientów. Okazuje się bowiem, że wyposażenie klientów w pełnowartościowe komputery (np. PC) powoduje szereg problemów, np. konieczność instalowania oprogramowania na wielu różnie skonfigurowanych komputerach, konieczność częstej wymiany sprzętu wraz ze wzrostem wymagań systemów, problemy z bezpieczeństwem (np. tworzenie przez użytkowników niezabezpieczonych lokalnych kopii danych, zagrożenie przez wirusy komputerowe), brak pełnej kontroli nad działaniami pracowników, brak kontroli licencji.

By uporać się z tymi problemami duże organizacje muszą zatrudniać rzesze administratorów, co powoduje, że koszt użytkowania systemów staje się wysoki.

Zaradzić tym kłopotom ma architektura trójwarstwowa. W architekturze tej wykorzystano właściwości *World Wide Web* i przeglądarek WWW.

Pierwszą warstwę stanowi uproszczony i tani komputer typu NC (*Network Computer*), nie mający własnego dysku ani stacji dyskietek, podłączony do sieci i wyposażony w przeglądarkę WWW (rolę takiego komputera może oczywiście pełnić także zwykły PC). Ta warstwa odpowiada głównie za prezentację, opartą na językach HTML i Java.

Drugą warstwą jest tzw. serwer aplikacji: komputer centralny, który realizuje logikę aplikacji (w tym logikę interfejsu użytkowników, w wyjątkiem samego wyświetlania) i część przetwarzania danych. Aplikacje wszystkich użytkowników są więc realizowane centralnie.

Trzecią warstwę stanowi serwer danych, na którym działa DBMS. Serwer aplikacji jest klientem w stosunku do serwera danych, zadania serwera danych są zatem takie same, jak w architekturze klient-serwer.

Architektura trójwarstwowa pozwala uniknąć problemów związanych z zarządzaniem stacjami klientów (musi tam działać jedynie standardowa przeglądarka WWW). Aplikacją i bazą danych można administrować centralnie. Wadą jest konieczność użycia silnego komputera na serwer aplikacji. Architektura ta przystosowana jest przede wszystkim do potrzeb dużych organizacji.

2.4. Rodzaje narzędzi dostępu do baz danych

Aplikacja jest tą częścią systemu informatycznego, z którą bezpośrednio styka się użytkownik i która najczęściej stanowi przedmiot krytyki. Musi więc być zaprojektowana i wykonana starannie i jednolicie, musi być przyjazna dla użytkownika i podatna na modyfikacje.

Typowe moduły aplikacji — formularze i raporty — tworzy się na ogół używając specjalizowanych narzędzi. Inne moduły wymagają programowania w bardziej uniwersalnych językach.

2.4.1 Narzędzia do tworzenia aplikacji

Aplikacje systemu informatycznego z bazami danych można tworzyć wieloma różnymi technikami, stosując różne klasy narzędzi.

Oto typowe podejścia do tworzenia aplikacji i cechy narzędzi do tego używanych:

- Tradycyjne programowanie w uniwersalnych językach trzeciej generacji (3GL), np. C, C++. Wymaga to zastosowania interfejsu między DBMS i językiem programowania, np. tzw. zanurzonego języka SQL (*embedded SQL*). Możliwości programowania aplikacji są niemal nieograniczone, ale jest to mozolne i kosztowne: wydajność programisty jest bardzo mała, a utrzymanie programów bardzo trudne. Podejście to jest odpowiednie jedynie do tworzenia nietypowych modułów, np. specjalizowanych połączeń między systemami i szybkich modułów przetwarzających dane.
- Programowanie wzbogacone o projektowanie graficzne za pomocą tzw. języków „wizualnych". Daje duże możliwości i znacznie większą wydajność, ale narzędzia są dość trudne w użyciu, najprostsze nawet działanie aplikacji wymaga bowiem programowania, a gotowe programy są dość trudne do utrzymania.
- Specjalizowane narzędzia czwartej generacji (4GL). Aplikacje projektuje się tu graficznie, a typowe działania aplikacji nie wymagają w ogóle programowania. Pozwala to osiągnąć dużą wydajność programisty, narzędzia są dość łatwe w użyciu, a gotowe aplikacje stosunkowo łatwe do utrzymania. Natomiast na ogół trudne jest uzyskanie nietypowego działania aplikacji. Gotowa aplikacja do działania wymaga zwykle instalacji specjalnego oprogramowania, tzw. *runtime*. Za to aplikacje zbudowane tą technologią często są przenośne, tzn. można je uruchamiać bez większych zmian na różnych platformach systemowych i sprzętowych.
- Automatyczne generowanie aplikacji. Aplikację projektuje się specyfikując jej cechy (np. w systemie CASE), a następnie generuje się ją automatycznie, na ogół jako gotowy program w języku 3GL lub 4GL. Projektant osiąga bardzo dużą wydajność, a programista bywa w ogóle niepotrzebny. Programy tak uzyskane są jednolite, przenośne i łatwe w utrzymaniu (w razie konieczności zmian modyfikuje się specyfikację i powtarza generowanie). Bardzo trudne bywa uzyskanie nietypowego działania aplikacji, a same narzędzia są na ogół drogie, obszerne i trudne do opanowania.
- Budowa systemu z komponentów. Jest to nowe, ale niezwykle obiecujące podejście do tworzenia systemów: system składa się z gotowych (wytworzonych wcześniej albo zakupionych) komponentów obiektowych. Obiekty-komponenty wchodzące w skład systemu świadczą sobie określone usługi. System taki może być rozproszony, tzn. komponenty mogą funkcjonować na wielu komputerach. Podstawę budowy takiego systemu stanowi uniwersalna warstwa komunikacyjna, przekazująca żądania usług i odpowiedzi między obiektami, nazywa się ona ORB (*Object Request Broker*). Najbardziej znanym standardem architektury takich systemów jest CORBA (*Common Object Request Broker Architecture*).

Sposoby tworzenia aplikacji, stosowane dla tradycyjnych architektur systemu informatycznego (przede wszystkim klient-serwer), mają swoje odpowiedniki dla architektury trójwarstwowej:

- Programowaniu w językach ogólnego przeznaczenia odpowiada pisanie programów typu CGI, udostępniających dane z DBMS przez WWW.
- Programowaniu w językach „wizualnych" odpowiada tworzenie aplikacji w języku Java w użyciem specjalnego interfejsu do baz danych, zwanego JDBC (*Java Database Connectivity*).
- Niektóre narzędzia czwartej generacji pozwalają bezpośrednio tworzyć aplikacje przeznaczone dla architektury trójwarstwowej.
- Istnieją także generatory generujące aplikacje w architekturze trójwarstwowej.

2.4.2 Narzędzia dostępu *ad hoc*

W każdym poważnym systemie z bazą danych niezbędne jest zastosowanie narzędzi dostępu *ad hoc* do danych. Wynika to z potrzeby dokonywania różnorodnych zapytań, których nie da się z góry przewidzieć w czasie tworzenia systemu (np. zapytań związanych z analizą danych, zmieniającą się sprawozdawczością itp.).

Narzędzia dostępu *ad hoc* należą zwykle do jednej z poniższych grup:

- *Konsole języka SQL* — są to uniwersalne programy pozwalające interaktywnie zadawać zapytania w języku SQL. Praktycznie każdy RDBMS jest wyposażony w tego typu narzędzie. Konsola SQL umożliwia zadanie każdego poprawnego zapytania. Oczywiście do użycia konsoli wymagana jest znajomość SQL, narzędzia z tej grupy są więc przeznaczone głównie dla informatyków. Zwykle jest możliwe przygotowanie skryptów zawierających zapytania w SQL; takie gotowe skrypty mogą być oczywiście używane przez mniej wykwalifikowany personel.
- Programy do „graficznego" tworzenia zapytań, typu QBE (*Query By Example*) — stanowią najliczniejszą grupę narzędzi do zapytań *ad hoc*. Zapytanie buduje się tu składając je z elementów wybieranych z listy, związki i warunki ustanawia się za pomocą symboli graficznych. Bardziej wyrafinowane narzędzia tego typu pozwalają administratorowi utworzyć specjalną warstwę pośrednią, prezentującą użytkownikowi dane już częściowo przygotowane do prezentacji, np. połączone w typowe związki. Narzędzia z tej grupy są z reguły łatwe w obsłudze i nie wymagają od użytkownika znajomości języka SQL, choć do ich efektywnego użycia niezbędne jest zrozumienie podstawowych zasad budowy i działania relacyjnych baz danych.
- Programy do interaktywnej analizy danych — są to narzędzia podobne do opisanych poprzednio, ale bardziej wyspecjalizowane: umożliwiają nie tylko zadanie zapytania i obejrzenie jego wyników w prostej formie, ale także analizę tych wyników (najczęściej analizę wielowymiarową, co opisano dalej).
- Interfejsy dostępu do danych z programów biurowych — pozwalają na import danych z bazy danych do programów biurowych (edytorów tekstu, arkuszy kalkulacyjnych itp.). Standardowym interfejsem stosowanym do tego typu połączeń jest ODBC (*Open Database Connectivity*). Do zadawania zapytań do bazy można tu użyć języka SQL lub narzędzia typu QBE, stanowiącego część pakietu biurowego.

3. Hurtownie danych

W dużych współczesnych organizacjach pojawiła się potrzeba analizowania danych dotyczących bieżącej i przeszłej działalności przedsiębiorstwa. Analiza taka może stanowić mocną podstawę do podejmowania decyzji dotyczących zarządzania przedsiębiorstwem.

Okazało się jednak, że istniejące w organizacjach systemy informatyczne nie mogą dostarczyć potrzebnych danych. Dane zgromadzone w istniejących dotychczas bazach danych (zwanych bazami operacyjnymi) są bowiem rozproszone, niejednorodne, a systemy informatyczne często nie są zintegrowane ani nawet połączone. Systemy informatyczne w organizacji pochodzą zwykle od wielu producentów, przechowują dane w różnych bazach danych i systemach zapisu, dane są w różnych formatach, a ich budowa opierać się może na różnych modelach danych. Układ

danych jest dostosowany do potrzeb operacyjnych, dane są więc przechowywane w sposób umożliwiający ich efektywną modyfikację, a układ taki z reguły nie sprzyja sprawnej analizie. W operacyjnych bazach danych przechowuje się na ogół dane odzwierciedlające jedynie stan aktualny lub najnowszą historię, tymczasem do analiz i porównań mogą być potrzebne długookresowe dane historyczne.

W tej sytuacji konieczne stało się znalezienie specjalnych rozwiązań, umożliwiających efektywną analizę danych zgromadzonych w organizacji. Rozwiązaniem, które zdobyło największą popularność, jest hurtownia danych.

Hurtownia danych (magazyn danych, *data warehouse*) jest wydzieloną centralną bazą danych, zbierającą informacje służące do zarządzania organizacją. Baza ta jest odizolowana od baz operacyjnych, a jej struktura i użyte do jej budowy narzędzia powinny być zoptymalizowane pod kątem przetwarzania analitycznego. W hurtowni są gromadzone dane pozyskiwane okresowo z systemów obsługujących dane operacyjne.

Ponieważ w hurtowniach gromadzi się dane z wielu źródeł i przechowuje się nie tylko dane bieżące, ale i historyczne, hurtownie mają z reguły wielkie rozmiary.

3.1 Przetwarzanie transakcyjne a przetwarzanie analityczne

Typowe systemy informatyczne, działające dotychczas w organizacjach, są systemami przetwarzania transakcji (OLTP — *On Line Transaction Processing*). Zadaniem takich systemów jest wydajne obsługiwanie zmian danych, wprowadzanych jednocześnie przez wielu użytkowników.

Najczęściej systemy OLTP sprawnie obsługują wielką liczbę stosunkowo niewielkich transakcji wykonywanych równocześnie. Tymczasem w przetwarzaniu analitycznym wielodostęp i efektywna modyfikacja danych są problemami drugorzędnymi. Istotna jest natomiast efektywność operacji wyszukiwania, odczytu i agregowania bardzo dużych objętości danych.

Dlatego systemy analityczne (OLAP — *On Line Analytical Processing*) muszą mieć odmienną konstrukcję, a do ich budowy używa się specjalnych narzędzi, a często nawet specjalizowanych systemów zarządzania bazami danych.

3.2. Analiza wielowymiarowa

Najbardziej typowym zadaniem systemu OLAP jest analiza wielowymiarowa. Dane gromadzone w hurtowniach danych najczęściej mają bowiem właśnie charakter wielowymiarowy — umożliwiają prowadzenie analiz wpływu wielu różnych czynników na zjawiska zachodzące w organizacji.

Struktura wielowymiarowa przedstawia elementarne komórki danych, tzw. fakty, w funkcji wielu niezależnych czynników, zwanych wymiarami.

Wymiary są opisane wartościami dyskretnymi, które mogą tworzyć hierarchie. Typowe wymiary to np. czas (np. w dniach, miesiącach, kwartałach, latach), produkt (np. typ i rodzaj), jednostka organizacyjna (np. wydział, oddział) lub terytorialna (np. gminna, powiatowa, wojewódzka).

Fakty są opisane atrybutami liczbowymi, tzw. miarami. Najbardziej typowym faktem jest sprzedaż, której miarami są np. ilość sprzedanego towaru i jego wartość.

W czasie analizy dane wielowymiarowe są poddawane pewnym typowym operacjom, takim jak obracanie, selekcja (wybór interesujących elementów wymiarów), projekcja (zmniejszenie liczby wymiarów, dane zostają zagregowane względem usuniętych wymiarów), wycinanie (*slice and dice* — połączenie selekcji i projekcji), ranking (uszeregowanie elementów wymiaru wg wzrostu miary lub jej agregatu), zwijanie i rozwijanie (*roll-up* i *drill-down* — nawigacja po hierarchii wymiaru, połączona, odpowiednio, z agregacją lub dezagregacją miar).

Celem tych wszystkich operacji jest zwykle dogodna prezentacja danych na płaszczyźnie dwuwymiarowej, w takiej „perspektywie", jaka potrzebna jest analizującemu.

3.3 Architektury hurtowni danych

Centralna hurtownia danych gromadzi dane niezbędne do tworzenia wszelkich potrzebnych analiz. Oprócz danych elementarnych, czyli kopii danych źródłowych, hurtownia przechowuje także tzw. zmaterializowane agregaty — zapamiętane wyniki obliczeń (np. sumy); pozwala to przyspieszyć analizę danych. Okazuje się jednak, że pożądany sposób agregacji danych silnie zależy od rodzaju prowadzonych analiz. Poszczególne części organizacji mogą potrzebować różnych danych, zagregowanych w odmienny sposób. Dlatego tworzy się mniejsze, wyspecjalizowane zbiory danych, zwane składnicami danych (*data marts*), zwykle przeznaczone dla wydziałów organizacji.

Centralna hurtownia danych (*data warehouse*) jest uniwersalna i gromadzi dane elementarne pokrywające potrzeby wszystkich przewidywanych analiz. Jest to ona przeznaczona do wykorzystania w całej organizacji. Zawiera dane historyczne i przechowuje dane słabo zagregowane. Dane w centralnej hurtowni pochodzą zwykle z wielu źródeł operacyjnych.

Składnice danych (*data marts*) są specyficzne dla zastosowania — ich budowa jest inna w każdej z części organizacji, dostosowana do prowadzonych tam analiz. Dane w różnych składnicach mogą się powtarzać. Dane są zwykle silnie zagregowane i zdenormalizowane (tzn. zawierają powtórzenia), struktura danych jest bowiem zoptymalizowana pod kątem efektywności lokalnie prowadzonych analiz. Najczęściej dane w składnicach pochodzą tylko z jednego źródła — z centralnej hurtowni danych.

3.4. Technologie dla hurtowni danych

Hurtownie danych gromadzą wielkie ilości danych, a ich przeznaczeniem jest przetwarzanie analityczne, które musi być dostatecznie wydajne. Dlatego do budowy hurtowni danych i aplikacji analitycznych potrzebne są specjalne technologie informatyczne, dostosowane do tych specyficznych potrzeb.

3.4.1 DBMS dla hurtowni danych

Fakty stanowią centralny punkt struktury danych wielowymiarowych, są one powiązane związkami z wymiarami, które z kolei tworzą hierarchie. Struktura danych przybiera więc na ogół formę gwiaździstą. Reprezentacja takich danych w relacyjnych bazach danych prowadzi do złożonych struktur, trudnych do efektywnego przetwarzania.

Typowe relacyjne bazy danych mogą więc nie podołać wymaganiom wynikającym z przetwarzania analitycznego, przede wszystkim z uwagi na ograniczoną wydajność realizacji operacji typowych dla analizy wielowymiarowej. Dlatego obok systemów zbudowanych z użyciem relacyjnych baz danych (tzw. ROLAP — *Relational OLAP*) stosuje się specjalnie skonstruowane serwery wielowymiarowe (MOLAP — *Multidimensional OLAP*). Systemy relacyjne są w stanie przechowywać ogromne ilości danych (terabajty). Serwery wielowymiarowe mogą wprawdzie pomieścić mniejsze objętości danych (gigabajty), ale są w stanie bardzo efektywnie obsługiwać zapytania analityczne. Dlatego do tworzenia wielkich centralnych hurtowni danych stosuje się często rozwiązania ROLAP, podczas gdy wydziałowe składnice danych tworzy się używając technologii MOLAP.

Producenci relacyjnych systemów zarządzania bazami danych dostarczają specjalne wersje lub opcje swoich serwerów danych, by osiągnąć zadowalającą wydajność przetwarzania danych w hurtowniach. Do typowych rozwiązań należą: indeksy bitowe — przyspieszające wyszukiwanie danych spełniających podane warunki, indeksy złączeniowe — przyspieszające łączenie danych z różnych tabel, specjalne procedury optymalizacji zapytań do struktur gwiaździstych, inteligentne partycjonowanie — umożliwiające rozsądne podzielenie danych na wiele dysków i przeszukiwanie tylko tych części, które są potrzebne. Mniej typowe rozwiązanie stanowi zapisywanie danych w bazie nie wierszami (jak to się zwykle robi), lecz kolumnami.

3.4.2. Narzędzia do budowy aplikacji analitycznych

Przetwarzanie analityczne musi zapewnić efektywne analizowanie wielkiej ilości danych. Prezentacja danych nie może zależeć od sposobu ich przechowywania, ale musi być przejrzysta i zrozumiała dla użytkownika. Przeznaczeniem narzędzi analitycznych nie jest wykonanie aplikacji, która będzie użytkowana bez większych zmian, lecz interaktywne tworzenie stale zmieniających się analiz. Budowanie różnorodnych form prezentacji wyników analizy musi więc być łatwe i szybkie. Narzędzia muszą być szczególnie wygodne i przyjazne, gdyż ich użytkownikami nie mają być programiści, lecz nie mający przygotowania informatycznego specjaliści z dziedziny zarządzania.

Dlatego do budowania aplikacji typu OLAP nie używa się na ogół typowych narzędzi do tworzenia aplikacji. Stosuje się wydajne narzędzia specjalizowane, łatwe w obsłudze i posiadające duże możliwości prezentacji danych, także w postaci graficznej. Wynikowe analizy zwykle dają się łatwo przenieść do dokumentów tworzonych typowymi programami biurowymi oraz do prezentacji w WWW. Niektóre narzędzia pozwalają budować analizy z przygotowanych wcześniej składników. Na rynku spotyka się także gotowe aplikacje analityczne, rozwiązujące typowe problemy, ale wyposażone w możliwości łatwego rozbudowywania o nowe analizy.

4. Podsumowanie

Podstawę konstrukcji wielkiej części profesjonalnych systemów informatycznych stanowią bazy danych. Pokazano, że jest to uzasadnione i celowe.

Technologie baz danych i systemów zarządzania bazami danych oraz tworzenia aplikacji z bazami danych stale się rozwijają. Mimo wyraźnej dominacji baz relacyjnych, architektury klient-serwer i narzędzi programistycznych czwartej generacji powstają nowe, obiecujące rozwiązania i tworzą się nowe, ważne trendy, które w najbliższych latach mogą zastąpić obecnie stosowane rozwiązania.

Jednym z ważniejszych, stosunkowo nowych trendów w rozwoju systemów informatycznych jest technologia hurtowni danych. Zdobywa ona wielkie powodzenie, ponieważ zaspokaja praktycznie ważne potrzeby wielkich organizacji.

Polecana literatura:

[1] W. Harris: *Bazy danych nie tylko dla ludzi biznesu.* WNT, 1994.

[2] K. Subieta: *Słownik często spotykanych terminów dotyczących obiektowości.* IPI PAN, 1997.

[3] http://www.ia.pw.edu.pl/~ttraczyk/bssi.html

[4] W.H. Inmon: *Building the Data Warehouse.* J. Wiley, 1992.

[5] T. Traczyk: Hurtownie danych. Materiały konferencji Centrum Promocji Informatyki *Bazy danych.* Kazimierz Dolny, kwiecień 1998.

Informacja o autorze:

dr inż. Tomasz Traczyk jest adiunktem w Instytucie Automatyki i Informatyki Stosowanej na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej

e-mail: ttraczyk@ia.pw.edu.pl

URL: http://www.ia.pw.edu.pl/~ttraczyk/

Grzegorz Janczewski

THE DOCUMENT COMPANY XEROX

Xerox Polska Sp. z o.o.
00-762 Warszawa
ul. Belwederska 20/22
Telefon: /22/ 651 18 45; 646 52 50
Fax: /22/ 651 18 46

Nowa strategia XEROX – technologia cyfrowa od biurka do masowej produkcji dokumentów

Coraz szersze stosowanie technologii cyfrowej we wszystkich dziedzinach pracy nie ominęło również produkcji dokumentów. Technologia ta umożliwia połączenie wielu funkcji w pojedynczym urządzeniu i pozwala na połączenie nowoczesnych mediów z wciąż najpopularniejszym papierem.

Xerox Corporation – światowy lider rynku przetwarzania dokumentów, realizuje strategię pełnego wykorzystania możliwości oferowanych przez technologię cyfrową. Najnowszym jej elementem jest wprowadzenie urządzeń, z którymi coraz częściej będą mieli do czynienia ludzie tworzący strategie zastosowania rozwiązań informatycznych we wszelkiego rodzaju organizacjach. Urządzenia te pozwalają wykonywać kopie, drukować, przesyłać faksy czy skanować dokumenty w sieciach lokalnych. Ich wydajności są dostosowane do każdych potrzeb. Od najmniejszych, jednoosobowych biur, po centra obliczeniowe lub działy poligrafii wielkich koncernów.

Drugim elementem tej strategii jest wyraźne rozgraniczenie środowisk pracy biurowej i profesjonalnej produkcji. Jako jedyna firma na świecie, Xerox proponuje odpowiednie rozwiązania dla każdego segmentu rynku. Bazując na osiągnięciach własnych działów badawczych i rozwojowych opracowano systemy, które mogą, w zależności od konfiguracji pracować jako kopiarko-drukarko-skanery w biurach lub być wykorzystywane w połączeniu z profesjonalnymi systemami archiwizacji i skanowania do drukowania publikacji takich jak raporty, instrukcje obsługi, niskonakładowe książki itp. Inna konfiguracja tego samego urządzenia umożliwia sprawną realizację druku faktur, wyciągów czy innych dokumentów transakcyjnych.

Podczas 14 Jesiennych Spotkań PTI w Mrągowie Xerox Polska przedstawi elementy tej strategii i zaprezentuje rozwiązania, z którymi coraz częściej będą mieli do czynienia pracownicy informatyki, przy podejmowaniu decyzji o metodach produkcji dokumentów. O wadze tych decyzji może świadczyć fakt, że według niezależnych szacunków nawet do 10 % budżetów firm może być wydawane właśnie na produkcję dokumentów. Wybór właściwej strategii staje się więc coraz bardziej decyzją o znaczącej wadze ekonomicznej, a znajomość najnowszych tendencji i rozwiązań z pewnością ułatwia podejmowanie takich decyzji.

Jakub Połeć

Microsoft Sp. z o.o.
Ul. Grzybowska 80/82
00-844 Warszawa
tel. (0-22) 661-54-00
fax (0-22) 661-54-34

Windows NT 4.0 Terminal Serwer Edition

Windows NT 4.0 Terminal Server Edition

Jakub Połeć
Microsoft

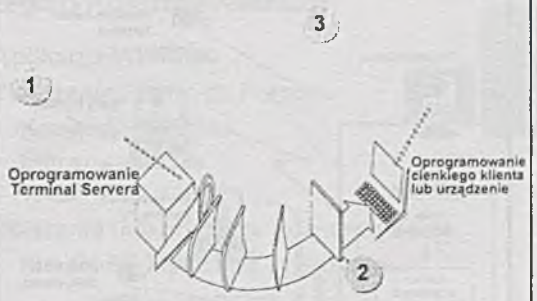
Plan

- Zapotrzebowanie
- Składniki Systemu
- Skalowalność
- Przewaga nad konkurencją
- Terminal Server a Citrix
- Instalacja Aplikacji
- Przyszłość

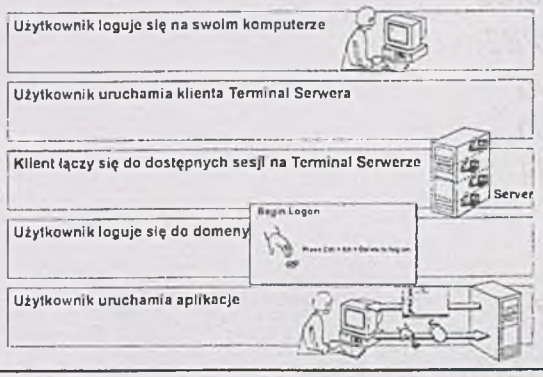
Windows NT Terminal Server

- Most łączący pulpity Win32 każdego użytkownika niezależnie od miejsca w którym on pracuje
  - Możliwość zastąpienia starszych wersji Terminali nowoczesnymi Terminalami Windows

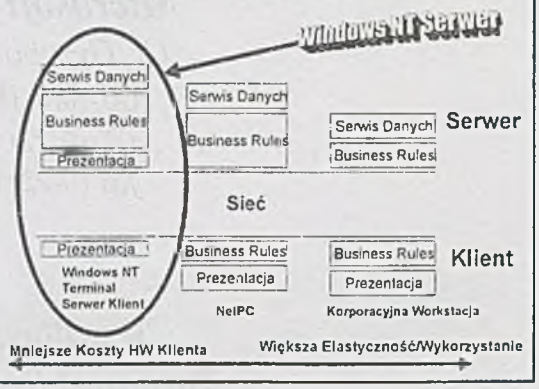
Zasada Działania Serwera



Zasada Działania Klienta



Windows Klient-Serwer



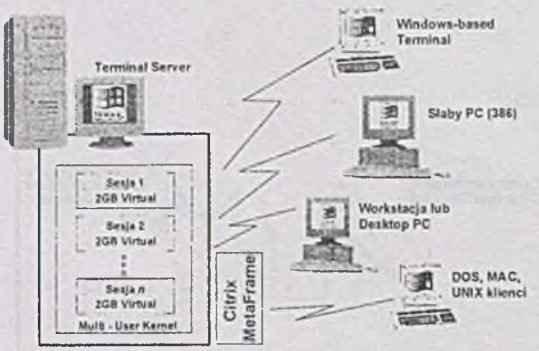
Windows NT Terminal Server

Architektura Klient-Serwer

- Wykorzystanie posiadanego sprzętu (słabszego, przestarzałego...)
- Centralna administracja - wszystko dzieje się na serwerze
- Bezpieczeństwo - wszystko na serwerze
- Praca przez wolne łącza - przesyła się tylko ekran
- Dostęp do konsoli z dowolnego miejsca

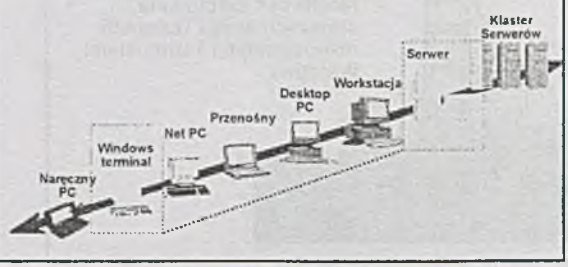
Rodzina Produktów Windows

Klient Terminal Server



Windows NT Terminal Server

- Poszerzenie i zintegrowanie rodziny produktów Windows
- Łatwy dostęp do wysoko wydajnego środowiska z relatywnie tanich klientów



Windows Terminal

- Jądro Systemu Windows CE w ROM
  - boot i połączenie do Windows Terminal Serwera
- Zawiera protokół RDP
  - Udostępnia Win32 UI (Windows 95 UI)
  - RDP protokół pochodzący od T.Share/T.120
  - OEMy mogą zapewnić dodatkową emulację terminala (ICA, 5250, 3270, VT100, etc.)
- Wsparcie dla przeglądarek i wywoływania aplikacji Javy nie jest związane z serwerem
- Urządzenia ICA skłaniają się w kierunku WBT

Wymiana Terminali

- Niska Cena Sprzętu
- Mniej niż $500 za Pojedynczego Klienta
- Sprzedaż przez OEM
  - tradycyjni sprzedawcy terminali
  - tradycyjni sprzedawcy Windows PC OEM
  - Działa na "super cienkich" klientach Windows

Net PC versus WBT

	Net PC	WBT
Lokalne wykonywanie aplikacji	Tak	Nie, wszystko wykonywane jest na serwerze
Dostęp do aplikacji Win32	Tak	Tak
Lokalna przeglądarka lub Java VM	Tak	Nie
Możliwość pracy w trybie offline	Tak	Nie
Elastyczność	Tak	Minimalna

Plan

- Zapotrzebowanie
- **Składniki Systemu**
- Skalowalność
- Przewaga nad konkurencją
- Terminal Server a Citrix
- Instalacja Aplikacji
- Przyszłość

Składniki Systemu
Trzy Podstawowe Składniki

- Przegląd/Kontrola Klientów
- Protokół Transferu Interfejsu Użytkownika ⇒ RDP
- Jądro pozwalające na jednoczesną pracę wielu użytkowników
  - Multi-User Object Manager
  - Virtual Memory Manager
  - Win32K.sys

Składniki Systemu
Przegląd/Kontrola Klientów

- Aplikacje Windows
- Tworzenie Zdalnych Pulpitów
  - Komendy Graficzne
  - Bitmapy
  - Tekst
- Zbieranie Informacji ze Zdalnych Sesji
  - Klawiatura, Mysz, Inne urządzenia

Składniki Systemu
Protokół Transferu Interfejsu Użytkownika

- Interpretuje i Przekazuje Wyniki z Pulpitu Użytkownika na Serwerze do Klienta
  - Kolejność Rysowania na Pulpicie
  - Bitmapy
- Transfer Informacji z Klawiatury i Myszki ze Zdalnego Klienta do Sesji na Serwerze
  - WM_KEY*, WM_MOUSEMOVE, etc.

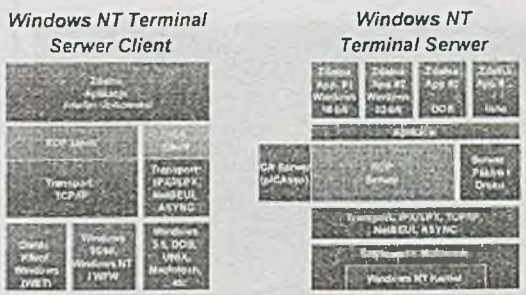
Składniki Systemu
Protokół Transferu Interfejsu Użytkownika

- Protokół RDP (rozszerzenie T.SHARE/T.120)
  - Międzynarodowy Standard Wielo-Kanałowego protokołu Konferencji
  - Używany przez Microsoft NetMeeting
  - Działa na TCP/IP
- Obecnie zoptymalizowany dla wysokoprzepustowych środowisk
- Wspiera szyfrowanie sesji
- Możliwe rozwiązanie przyszłościowych zastosowań audio/video

Składniki Systemu
Wieloużytkownikowe Jądro

- Otwarte Środowisko Serwera dla użytkownika sesji 0 i dla innych sesji
  - Zasoby komputerowe przydzielane i zarządzane są na poziomie użytkownika
    - Pamięć
    - CPU
    - Dysk i Inne Urządzenia
    - Usługi Podstawowe (np. OLE, Win32, Kernel object, ...)

Ostateczny Produkt
Architektura Klient-Serwer



Instalacja Windows NT Terminal Server

Windows NT Terminal Server
Instalacja

- Kompletny Setup (NT 4.0 + SP3)
- Wymagania
  - TCP/IP
  - Opcja by instalować IE 4.01
  - Możliwość dodania Licencji NTW dla Klientów
- Instalacja/Upgrade
  - Czysta Instalacja
  - Upgrade Serwerów WinFrame 1.6 lub 1.7
  - Brak NTS 4.0 Upgrade

Windows NT Terminal Server
Citrix WinFrame Upgrades

- Upgrade
  - WinFrame 1.6 i 1.7
  - OEM WinFrame (WinCenter, WinDD, NTrigue)
- Identyczny upgrade jak 3.51 -> 4.0
- Ustawienia ICA
  - Zachowane
  - Wymaga Citrix MetaFrame

Plan

- Zapotrzebowanie
- Składniki Systemu
- **Skalowalność**
- Przewaga nad konkurencją
- Terminal Server a Citrix
- Instalacja Aplikacji
- Przyszłość

Skalowalność

- 15 użytkowników na procesor
  - Pentium Pro: 20-25 użytkowników na serwer (UPS)
  - Win16: UPS mniejsze o 20%, RAM więcej o 25%
- 32 MB RAM pamięci w serwerze
  - 4-8 MB RAM dla każdego użytkownika
- wystarczający transfer przez sieć dla każdego użytkownika
  - 60 wymagających użytkowników = 1Mb/s
- Korzystanie z WWW
  - Internet Explorer: 50, Navigator: 25 UPS

Skalowalność

- Light user
  - Około ~40 użytkowników na CPU
  - Wykorzystanie sieci ~ 3 kbs każdy użytkownik
  - Wykorzystanie pamięci ~ 5 Mb każdy użytkownik
- Medium (Administrative) user
  - Około ~27 użytkowników na CPU
  - Wykorzystanie sieci ~5 kbs każdy użytkownik
  - Wykorzystanie pamięci ~ 8 Mb każdy użytkownik
- Heavy (Knowledge) User
  - Około ~15.5 użytkowników na CPU
  - Wykorzystanie sieci ~6 kbs każdy użytkownik
  - Wykorzystanie pamięci ~ 12 Mb każdy użytkownik

Ilu użytkowników ?

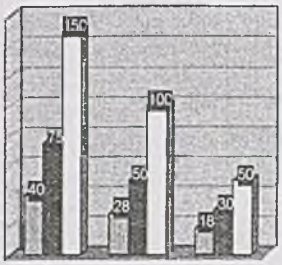
	Light User	Medium User	Heavy User
Dual PII 300 512 Mb RAM	90	60	37
DualPPro 512 Mb RAM	75	50	30
QuadPpro 1 Gb RAM	150	100	50

Skalowalność
Rekomendacje Compaq'a

UPS = Users per Server

- Heavy UPS:
  - Proliant 4500 1P-P5/166: 15 Heavy UPS
  - Proliant 5000 2P-P6/200: 30 Heavy UPS
  - Proliant 4500 3P-P5/166: 50 Heavy UPS
  - Proliant 4500 4P-P5/166: 70 Heavy UPS
  - Proliant 5000 4P-P6/200: 100 Heavy UPS
- Light UPS:
  - Prosignia 300 P5/90: 20 Light UPS
  - Proliant 4500 1P-P5/166: 40 Light UPS
  - Proliant 5000 2P-P6/200: 75 Light UPS
  - Proliant 5000 4P-P6/200: 200 Light UPS

Skalowalność
Rezultaty HP Scalability



Skalowalność
Rekomendacje Citrix'a

Maximum Concurrent Users	Processor	Memory
10 Users	Pentium 133	64Mb to 128Mb
45 Users	Pentium 133	128Mb to 160Mb
20 Users	Pentium Pro 200	192Mb to 224Mb
100 Users	Quad Pentium Pro 200	1Gb to 1.5Gb

- Dane z 12/16/97 ComputaCenter WinFrame Scalability & Application Guide whitepaper
- Rezultaty Testowe z:
  - WinFrame 1.6, 32 bit Office apps

Skalowalność
Wymagania dla Klientów

- Cel: uzyskanie „super cienkiego" klienta pozwalającego na pełną pracę w Środowisku Windows
- Wymagania Pamięciowe dla Klienta
  - ~500 kB na kod (wymagane)
  - ~100 kB na dane (wymagane)
  - Zmieniany cache dla bitmap (opcjonalnie)
  - Shadow screen bitmap (opcjonalnie)
  - Dla maszyny 800x600x256 około 1.1-1.5 MB

Skalowalność
Wymagania dla Klientów

- Mimimum VGA (640 x 480 16-color)
- Mysz + Klawiatura
- Połączenie sieciowe TCP/IP
  - Można używać poprzez RAS / Dial-Up
- Minimalne wymagania pamięci
  - WFW 3.11 4mb
  - Win 9x 8mb
  - Win NT 16mb

Plan

- Zapotrzebowanie
- Składniki Systemu
- Skalowalność
- **Przewaga nad konkurencją**
- Terminal Server a Citrix
- Instalacja Aplikacji
- Przyszłość

Przewaga nad konkurencją

- Pełna Znajomość Środowiska Windows
- Wsparcie Aplikacji
- Połączenie w Środowisku Heterogenenicznym
- Dostępność Aplikacji Win32 dla Różnorodnych Platform
- Łatwa Administracja

Mniejszy TCO

Przewaga nad konkurencją
Pełna Znajomość Środowiska Windows

- Znajomość środowiska Windows wśród użytkowników komputerów firmowych i domowych
- Ułatwienia w pełnej migracji do środowiska Win32
- Dostęp do wszystkich standardowych aplikacji Windows

Przewaga nad konkurencją
Wsparcie Aplikacji

- Kluczowe Aplikacje Mirosoft
  - Office, wersje 4.3c, 95, 97, Internet Explorer, wersje 3.x i 4.x, Exchange client, wersje 4.x i 5.x, Project v4.1a
- 16 i 32 bitowe Aplikacje Visual Basic
- Kluczowe Aplikacje Innych Firm
- Nowe Wieloużytkownikowe Wymagania opisane będą w produktach Windows i BackOffice
- Dokument "Jak Tworzyć Wieloużytkownikowe Aplikacje dla Windows"

Przewaga nad konkurencją
Testowane aplikacje

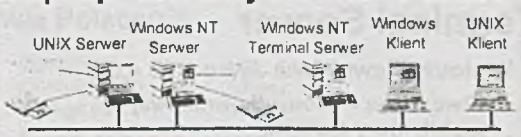
- MS Office 97
- MS Office 95
- MS Office 4.3
- SmartSuite 96
- Corel Office 8
- Corel Office 7
- Microsoft Project
- MS Access
- Outlook 98, 8.03
- IE 3.02, IE 4.01
- Navigator 3.0
- Communicator 4.0
- MS Visual Test
- Visual Basic Apps
- SNA Server 4.0
- Exchange Srv. 5.5
- MS SQL Srv. 6.5
- Lotus Notes Srv.

VeriTest - niezależna firma testująca oprogramowanie

Przewaga nad konkurencją
Połączenie w Heterogenenicznym Środowisku

- Łączenie z 32-bitowymi Aplikacjami Windows z:
  - Terminale X
  - Komputery MAC
  - Workstacje UNIX
  - Starsze komputery typu PC
- Łączność między aplikacjami dla mini- i mainframe'ów z Pulpitów Windows

Współpraca z systemami UNIX



- Cele Microsoftu :
  - Stworzyć najlepszą emulację terminala dla aplikacji Win32 w środowisku UNIXa
  - Zapewnić, że serwery UNIX'a i Windows NT będą współegzystowały wykorzystując nawzajem swoje silne strony
  - Dać użytkownikom swobodę w wyborze odpowiedniej platformy do pracy

Przewaga nad konkurencją
Aplikacje Win32 dla Różnorodnych Platform

- Klienci Windows NT Terminal Serwera
  - Terminale Windows (WBT)
  - Windows 95 i Windows NT Workstacja
  - Windows for Workgroups (Windows 3.11)
- Nie Windowsowi Klienci (via Citrix "pICAsso")
  - DOS, UNIX, MAC, NC

Przewaga nad konkurencją
Administracja

- Jednolita integracja z Rodziną Windows i ZAW
- Niskie koszty pozyskania nowych Terminali Windows
- Wybór Hardware (dla klienta i serwera)
- Uruchamianie standardowych, istniejących aplikacji
- Centralna administracja z narzędziami administracyjnymi Windows

Przewaga nad konkurencją
Mniejsze TCO

TCO ⇨ *Total Cost of Ownership*

- Niższy kapitał początkowy
- Niższe długoterminowe koszty zarządzania środowiskiem
- Niższe koszty szkoleń i wsparcia
- Niższe koszty tworzenia i instalacji aplikacji

Komputer Sieciowy
Niezgodne PC, Drogie Terminale...

- Model Komputerowy Windows NT Terminal Serwer
  - 100% server-based: nie ma potrzeby ładowania aplikacji do klienta
- Aplikacje Windows NT Terminal Serwer
  - 100,000-sięcy istniejących aplikacji Windows
  - 1000-iące narzędzi dla programistów i developerów programów Windows
  - Swobodny wybór metod programowania

Windows Terminal Klient

- Możliwość łączenia wcześniejszych sesji (disconnect)
- Wiele Terminali w obrębie jednej sesji
- Terminal w Terminalu
- Klient RAS over Windows Terminal Serwer

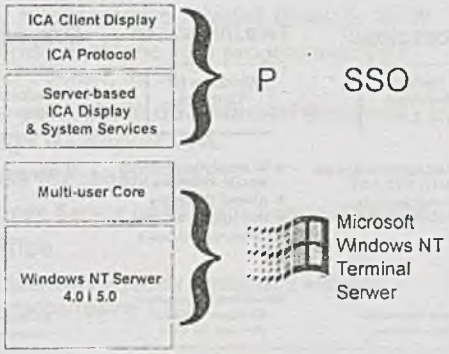
Plan

- Zapotrzebowanie
- Składniki Systemu
- Skalowalność
- Przewaga nad konkurencją
- **Terminal Serwer a Citrix**
- Instalacja Aplikacji
- Przyszłość

Citrix i NT Windows 4.0 Terminal Serwer

- Wielouzytkownikowe Jądro NTS 4.0
  - Stworzenie wielouzytkownikowej architektury serwera
- Nadzór w trakcie planowania i rozwoju
  - Wsparcie w stworzeniu wersji dla NT4 i NT5
  - Uczestnictwo w fazach Beta produktu
- Brak serwerów WinFrame dla NTS 4.0/5.0
  - WinFrame 1.6/1.7 do czasu wydania NTS
- Wsparcie dla istniejących i przyszłych klientów ICA via "pICAsso"

Projekt "pICAsso"



Plan

- Zapotrzebowanie
- Składniki Systemu
- Skalowalność
- Przewaga nad konkurencją
- Terminal Serwer a Citrix
- **Instalacja Aplikacji**
- Przyszłość

Windows NT Terminal Serwer

- NT Server 4.0 ze zmodyfikowanym jądrem
  - Praca w dwóch trybach w zgodnie z wymaganiami instalacji oprogramowania
- Podstawowe Wymagania HW
  - Intel Pentium
  - 32MB RAM, oraz
    - 4-8MB dla każdego typowego użytkownika
    - 16MB dla każdego więcej wymagającego
  - protokół TCP/IP

Windows NT Terminal Serwer
Narzędzia Administracyjne

- Terminal Server Administration
- Terminal Server Connection Configuration
- Rozszerzenia NT User Manger
- NetWare User Access for Terminal Server
- Application Security
- Nowe pola w Performance Monitor
- Security Configuration

Windows NT Terminal Serwer
Nowe Polecenia

change
change logon
change port
change user
cnvrtuc
connect
cprofile
dbgtrace
disconn
doskbd
flattemp
kill
logoff
migrate
msg
peruser
query
query object
query process
query session
query termserver
query user
register
reset
reset session
shadow
shutdown
wfprof

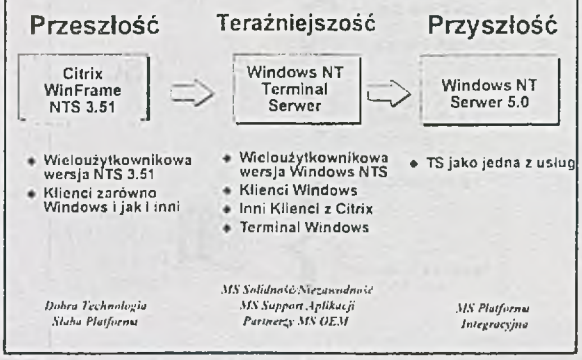
Windows NT Terminal Serwer Klient

- 16-bit - Windows for Workgroups
- 32-bit - Windows 95, Windows NT
- Wymagania
  - TCP/IP
  - RAM
    - 8MB Windows for Workgroups, Windows 95
    - 16MB Windows NT
  - 4MB przestrzeni dyskowej

Plan

- Zapotrzebowanie
- Składniki Systemu
- Skalowalność
- Przewaga nad konkurencją
- Terminal Serwer a Citrix
- Instalacja Aplikacji
- Przyszłość

Road Map



Przyszłość - Nowe Cechy

- Wiele kanałów dla pojedynczej sesji
- Clipboard
  - copy-paste między lokalnymi i zdalnymi aplikacjami
- Audio
  - kanały dźwięku i video
- Load Balancing
- Shadowing
- Web Client (ActiveX)Public APIs

Wersje Językowe

- 60 dni wersje: Hiszpańska, Niemiecka
- 60-90 dni wersje: Francuska, Japońska
- Wersja polska w wersji NT 5.0
  - uruchamianie zlokalizowanych aplikacji
  - polskie znaki na konsoli i u klienta

Spolszczenie NT 4.0 TSE

- Instalacja Option Pack 4.0 Polski
- Instalacja Internet Explorera 4.0 Polski
- Polski Office 97
- Polski Outlook 98

Licencjonowanie

Licencje

- Każdy klient musi posiadać licencję NTW
  - Licencje dostępne są w programach: FPP, MOLP, Select, itd..
- Każde połączenie do Terminal Serwera wymaga NT Server CAL
- License Manager Tool
  - License Server na każdym Terminal Server
- MS Office
  - Licencja na każdy terminal z którego uruchamiamy Office

Terminal Server
RDP Client Overview

- Presents Windows look n' feel
  - Renders all RDP protocol output
    - graphics, command, bitmaps, text, etc.
  - Gathers Keyboard / Mouse input
- Windows Application
- High Performance Thin Client
  - Multi-threaded (on Win32 platforms)
  - Intelligent bitmap and screen caches
  - Compact rendering core (~80K)

Terminal Server
RDP Client Overview

- Client Features
  - Simplified "Quick" connect UI w/ browser
  - Client Connection Manager Tool
  - Integration with Start Menu and Progman
- Supported Platforms
  - Windows 95 and 98, 8mb
  - Windows NT 3.51, 4.0, 5.0, 16mb
  - Windows for Workgroups (v3.11), 4mb
  - Windows-based Terminals (WinCE)

Terminal Server
Windows-based Terminals

- Dedicated Terminal Server client
- A limited function, low cost, and low maintenance device
- Runs thin RDP client software
- Sold by OEM to customer
- "Green screen" terminal replacement
- WinCE-based WBT OEM Kit
  - Core Windows CE OS
  - RDP client

Terminal Server
WBT Vendors

NCD
Tektronix
CruiseTechnologies™
Boundless
WYSE
neoware

Wojciech Sosnowski

Lotus Development Polska
Sp. z o.o.
ul. 1 Sierpnia 8
02-134 Warszawa
tel. 0-22/ 878-67-60
fax 0-22/878-67-64

Zarządzanie wiedzą przy użyciu narzędzi pracy grupowej

1. Knowledge Management Solutions

2. Formy zarządzania wiedzą

3. Strategia sukcesu w zarządzaniu wiedzą

4. Case studies:

- *Chase Manhattan Bank*
- *Philips North America*

Informacja vs Wiedza

Pomimo tego, że różnica pomiędzy informacją i wiedzą jest istotna, zdarza się często, że pomijamy ją przy tworzeniu systemów zarządzania wiedzą – a przy braku rozróżniania tych dwóch pojęć wartość systemu jest znacznie zmniejszona. Dyskusja na temat systemów zarządzania wiedzą powinna więc zaczynać się od podstawowego zrozumienia ogólnych pojęć takich jak wiedza, informacja i praca. Te trzy słowa opisują coraz bardziej skomplikowane działania, umożliwione przez rozwój technologii pracy grupowej.

Jak więc definiujemy różnicę pomiędzy informacją i wiedzą?

Na potrzeby niniejszej prezentacji, możemy uznać, że podstawą wszystkich trzech terminów są dane – proste, nieskategoryzowane fakty. Przykładem może być zapis z działu finansowego organizacji w rodzaju „Na Koncie A wystąpił debet w wysokości 5000 USD". Dane wzbogacone o kontekst stają się **informacją** – „Na koncie A, należącym do osoby XXX od 2 miesięcy brakuje 5000 USD".

Informację taką będziemy mogli interpretować jako **wiedzę**, w momencie, w którym będzie ona zawierać w kontekście odpowiednie „know-how" lub „know-why" – „wiedza-jak" oraz „wiedza-dlaczego" i dzięki temu wspierać proces decyzyjny. W naszym finansowym przykładzie, „know-how" mógłby wyglądać następująco: „Przed ogłoszeniem konta A za tzw „zły kredyt" (uznanie danych zobowiązań, kwot za niemożliwe do uzyskania), powinieneś zaoferować plan spłat. Jeżeli nie otrzymasz odpowiedzi, musisz zawiadomić przełożonego w celu otrzymania akceptacji unieważnienia długu". Jednocześnie „know-

why" może w tym przypadku stwierdzać, że „deklarujemy konta jako należące do „złych kredytów" wtedy, kiedy koszty odzyskania naszych pieniędzy będą większe niż same należności".

Pracę wreszcie, możemy definiować jako efekt wykorzystania wiedzy w działaniu – takim jak podjęcie decyzji o unieważnieniu długu i uznaniu go za tak zwany „zły kredyt".

Paweł Adamek, Mirosław Tarczoń

Océ-Poland Ltd. Sp. z o.o.
02-232 Warszawa, ul. Łopuszańska 53, tel. (0-22) 868 30 71, 868 30 76, 868 30 79
Internet: www.oce.com lub www.oce.com.pl

Kompleksowe rozwiązania w kopiowaniu i drukowaniu

Historia i profil firmy

Historia firmy sięga roku 1877, kiedy bracia van der Grinten rozpoczęli masową produkcję środków koloryzujących do masła i margaryny.

Szyld Océ (słowo utworzone od niemieckiego skrótu o.c.- ohne componenten, czyli bez dodatków) zastosowano po raz pierwszy w 1918 roku, kiedy bracia van der Grinten wynaleźli ulepszony niebieski papier do kopiowania techniką "blueprint". Niedługo potem nazwa ta stała się znana dzięki ich nowemu wynalazkowi - światłoczułej technologii "semi dry diazo". Patenty braci van der Grinten pozostały najpopularniejszą metodą kopiowania aż do lat 80-tych naszego wieku.

Firma rozwijała się niezwykle dynamicznie mimo rosnącej konkurencji i załamania gospodarczego związanego z obiema wojnami światowymi. W 1956r. firma Océ została wprowadzona na amsterdamską giełdę. Było to wyjątkowo udane posunięcie, bowiem lata 60-te przyniosły prawdziwy rozwój technik biurowych, a co za tym idzie, rosnące zapotrzebowanie na kopiarki dokumentów o wysokiej jakości.

Pod koniec lat 60-tych opracowano nową technologię kopiowania elektrostatycznego. W roku 1973 sprytni Holendrzy skonstruowali własną kopiarkę elektrostatyczną, wykorzystującą jednak technologię odmienną od kserograficznej. W ten sposób zapewnili sobie silną pozycję w Europie.

W 1978 roku Océ przejęło kontrolę nad swoim największym konkurentem brytyjskim koncernem Ozalid Group Holdings Ltd, działającym wtedy w 15 krajach. Zamiana rodzinnego interesu na nowoczesną spółkę akcyjną zaowocowała lawinowym rozrostem oraz ekspansją za granicą.

Podczas gdy na początku XX wieku Océ zatrudniało zaledwie kilku pracowników, już w 1966r. ich liczba wynosiła 2300, a firma posiadała 5 filii zagranicznych. Pozyskanie grupy Ozalid było jednym z wielu kroków na drodze do światowego sukcesu. Jego miarą było także wykupienie w roku 1996 działu drukarek wysokonakładowych firmy Siemens

Nixdorf, dzięki czemu obroty Océ wzrosły o jedną trzecią w skali roku oraz poszerzony został zakres oferowanych produktów.

Na dzień dzisiejszy obrót Océ sięga 3 mld dolarów amerykańskich rocznie, a zatrudnienie kształtuje się na poziomie 20.000 osób. Firma ma swoje przedstawicielstwa na wszystkich kontynentach, w ponad 90 krajach.

Obecnie Océ zajmuje pierwszą pozycję na świecie w produkcji urządzeń wielkoformatowych, drugą pozycję w profesjonalnym drukowaniu wysokonakładowym oraz należy do czołówki największych dostawców urządzeń do kopiowania wysokonakładowego.

Od początku swojej działalności rodzina van der Grinten inwestowała w wynalazczość i opracowywała nowe patenty. Także obecnie, co roku 7% obrotów przeznacza się na prace badawcze. Efektem jest ponad 1000 opatentowanych rozwiązań technicznych, m.in. metoda Copy Press polegająca na "wprasowywaniu" toneru w papier, nie zaś na "rozlewaniu" go w wysokiej temperaturze.

Copy Press zastosowano między innymi w modelu Océ 3045, który należy do tzw. systemów biurowych. Systemy te obejmują urządzenia do kopiowania (analogowego i cyfrowego) dokumentacji w formatach od A5 do A3. Wykorzystywana technologia Copy Press gwarantuje najwyższą jakość odwzorowania, bez smug z zabrudzonych elektrod i przesunięć toneru. Wśród systemów biurowych firma oferuje także urządzenia do przetwarzania informacji w kolorze. Océ zapewnia, aby stosowanie koloru zostało osadzone w całej infrastrukturze informatycznej oraz w systemie opracowywania i obiegu dokumentacji. Krótko mówiąc Océ oferuje oryginalne i kompleksowe podejście do stosowania barw i technologii cyfrowej w nowoczesnych przedsiębiorstwach.

Drugim obszarem działalności Océ są systemy wielkoformatowe; zaliczają się do nich kopiarki, plotery i skanery do tworzenia i powielania dokumentacji technicznej o wielkości do A0 lub większej. Największym jak dotąd osiągnięciem technologicznym firmy jest Océ 9800 - wyprodukowana w 1995r. unikatowa, wysokonakładowa cyfrowa kopiarko-drukarka. Oprócz opisanej już techniki Copy Press, dysponuje ona funkcją automatycznie eliminującą zakłócenia i wzmacniającą obraz - Image Enhancement. Jest to sprzęt najnowszej generacji, robiony na zamówienie i cieszący się olbrzymim uznaniem.

W zakresie drukowania oferta koncernu obejmuje systemy drukujące, w które wchodzą między innymi średnio- i wysokonakładowe drukarki. Paleta oferowanego sprzętu zawiera urządzenia o zakresie prędkości od 30 do 880 stron na minutę. Pod koniec 1997 roku wprowadzona została na rynek najszybsza maszyna z serii Pagestream, TWIN PS 1000. W tym samym roku na rynku znalazły się też wersje kolorowe systemów drukujących 200 DSC oraz wszystkich systemów Twin. Użycie kolorowych tonerów umożliwia dwustronne drukowanie z dodatkowym kolorem drukowanym na jednej stronie. Istnieje także możliwość wyboru jeszcze jednego dodatkowego koloru. Wypuszczono również na rynek serię maszyn pod nazwą Demandstream, zbudowanych na podstawie Pagestream. Dzięki specjalnemu tonerowi oraz rozdzielczości 600 dpi znajdują one zastosowanie w Printing on Demand: są to niewielkie nakłady dokumentów, których treść jest często zmieniana.

Systemy drukujące są podłączalne do różnych platform sprzętowych oraz akceptują wiele języków opisu strony jak również umożliwiają druk kodów paskowych, OCR i innych.

Firma Océ zapewnia również najwyższej jakości materiały eksploatacyjne dla swoich maszyn tj. papier, tonery, folie, kalki.

Océ w Polsce

W Polsce Océ działa z dużym powodzeniem od 1995 roku. Ma swoje oddziały już w siedmiu miastach - Warszawie, Krakowie, Wrocławiu, Poznaniu, Gdańsku, Katowicach i Szczecinie.

Misją firmy jest kompleksowa obsługa Klientów obejmująca dostarczanie urządzeń, materiałów eksploatacyjnych, konsultacji, serwisu, doradztwa technologicznego opartego na najnowocześniejszych, cyfrowych

rozwiązaniach technologicznych i najwyższym poziomie jakości. Zgodnie z założeniami misji firmy, Océ w Polsce oferuje dogodne warunki kupna, leasingu i dzierżawy urządzeń oraz zapewnia kompletny serwis i "opiekę" nad urządzeniami. Firma Océ ma swoich Klientów w najwyższych urzędach państwowych - Kancelarii Sejmu, Senatu, Ministerstwie Finansów oraz w wielkich firmach jak Unilever, Henkel, ABB czy Daewoo. Właśnie doskonała współpraca z klientami umożliwiła Océ Poland imponujący rozwój - w 1997 roku sprzedaż wzrosła o blisko 75% w stosunku do roku ubiegłego. Wielokrotnie wzrosła też sprzedaż urządzeń należących do tzw. systemów drukujących. Niezawodność i dobra jakość urządzeń została potwierdzona na Targach Poznańskich Infosystem i Poligrafia, gdzie w 1997 roku Złoty Medal otrzymało urządzenie wielkoformatowe Océ 9400, a w 1998 roku również Złoty Medal otrzymała cyfrowa kopiarko-drukarka. Nie można jednak zapominać, że Océ Poland jest częścią wielkiego, międzynarodowego koncernu, zajmującego pierwszą pozycję na świecie w produkcji urządzeń wielkoformatowych, drugą pozycję w profesjonalnym drukowaniu wysokonakładowym. Firma należy do trzech największych dostawców urządzeń do kopiowania wysokonakładowego. To zobowiązuje.

Ochrona środowiska

Naczelną zasadą filozofii Océ jest odpowiedzialność w stosunku do ludzi i środowiska. Na długo przed szerokim rozpowszechnieniem haseł ochrony środowiska, koncern projektował systemy i urządzenia przyjazne dla otoczenia człowieka.

Wieloletnie badania, w które koncern Océ inwestuje swoje fundusze, zaowocowały licznymi rozwiązaniami proekologicznymi. Zostały one opracowane przede wszystkim z myślą o bezpieczeństwie i wygodzie przyszłych użytkowników urządzeń Océ.

Maszyny naszej firmy charakteryzują się wyjątkowo niską emisją ozonu i pyłów. Innowacje rozwiązań spowodowały, że ilość wydzielanego przez nie ozonu jest 4÷8 razy mniejsza niż w przypadku innych kopiarek. Wiele substancji szkodliwych zastąpiono składnikami organicznymi. Inne związki są skutecznie odfiltrowywane z powietrza. W latach dwudziestych Océ wynalazło i opatentowało technologie kopiowania na materiałach światłoczułych bez użycia par amoniaku; w latach siedemdziesiątych rozpoczęło produkcję urządzeń kopiujących na zwykłym papierze, a obecnie Océ wytwarza kopiarki wykorzystujące tlenek cynku jako pokrycie fotoreceptora.

Dla użytkowników sprzętu Océ ważne jest również to, że naszym inżynierom udało się skonstruować maszyny nieporównywalnie ciche. Także niewielka ilość wydzielanego przez nie ciepła zapewnia komfort pracy.

Dobro środowiska naturalnego przyświecało autorom rozwiązań ukierunkowanych na oszczędność. I tak np. technika one-pass czy double-sided copying umożliwia oszczędność w zużyciu papieru. Nie bez znaczenia jest tu także wysoka jakość i niezawodność urządzeń Océ, które przyczyniają się do redukcji ilości kopii zmarnowanych.

Wszelkie wymogi związane z ochroną środowiska naturalnego spełniane są również na etapie produkcji urządzeń Océ. Dotyczy to między innymi oszczędnego zużycia energii i wody, a także materiałów do opakowań. Ponadto przy produkcji sprzętu tej firmy nie stosuje się substancji toksycznych przyczyniających się do redukcji warstwy ozonowej.

Urządzenia Océ są regenerowane i przerabiane. Możliwy jest recycling nawet części plastikowych i toneru. To, co nie poddaje się przeróbce, usuwane jest zgodnie z zasadami ustalonymi w programie ISO 9000, w sposób najbardziej korzystny dla środowiska.

W uznaniu dla proekologicznej polityki firmy, urządzenia Océ otrzymały szereg certyfikatów takich jak np. „Blue Angel", „Energy Star" czy „ISO 9002 Management System".

Wyrazem troski koncernu Océ o stan środowiska naturalnego jest również ustanowienie nagrody im. Océ-van der Grinten. Od 1997 roku jest ona przyznawana naukowcom z całej Europy za nowatorskie projekty proekologiczne.

Pierre Sabloniere

IBM Polska Sp. z o.o.
ul. 1 Sierpnia 8
02-134 Warszawa
tel. (0-22) 878-67-77
fax (-0-22) 878-68-80

Systemy ERP na platformie IBM S/390
IBM S/390 – Konsolidacja Serwerów

Korporacja IBM jest obecnie największą firmą informatyczną na świecie. Swoją pozycję budowała przez prawie 100 lat istnienia, w ciągu których zasady wprowadzone przez jej założyciela Thomasa J. Watsona - ZADOWOLENIE KLIENTA, SZACUNEK DLA JEDNOSTKI i DĄŻENIE DO DOSKONAŁOŚCI - były podstawą jej działania. Dziś IBM zatrudnia 240 615 ludzi w 164 krajach.

Na świecie znajduje się 13 ośrodków IBM w obrębie 4 geograficznych grup handlowo-usługowych:

* IBM United States i IBM Canada w Ameryce Północnej;
* IBM Latin America w Meksyku, Ameryce Środkowej i Południowej oraz na Karaibach;
* IBM Asia Pacific w Japonii i na Dalekim Wschodzie, w Australii i Nowej Zelandii;
* IBM Europe/Middle East/Africa (EMEA) w Europie, na Bliskim Wschodzie i w Afryce.

IBM był pierwszą firmą na świecie, która przeznaczała znaczne sumy na własną działalność badawczo-rozwojową. W Europie, w 9 krajach, istnieje 7 laboratoriów badawczych, w których około 5 tysięcy fachowców pracuje nad udoskonalaniem produktów, procesów produkcyjnych i programów komputerowych. Do IBM należy wiele wynalazków technologicznych kształtujących przemysł komputerowy, a naukowcy z laboratoriów IBM pięciokrotnie otrzymywali Nagrody Nobla. Spośród pięciu laureatów tego wyróżnienia trzech nadal pracuje w laboratoriach IBM. W ostatnich latach IBM opatentował więcej wynalazków niż jakakolwiek firma na świecie. Okazuje się, że najważniejsze Watsonowskie hasło – MYŚL! - nie straciło swej mocy.

Prawie wszystko co jest ważne w informatyce zrodziło się w ośrodkach badawczych IBM. Obecnie firma bardzo dynamicznie wprowadza własne technologie do produkcji oraz prowadzi podobne działania na zewnątrz, głównie poprzez licencje i część sprzedaży tak zwanego oryginalnego wytwórcy. Wpływy ze sprzedaży licencji

i praw autorskich niemal podwoiły się w ciągu ostatnich lat. Sprzedaż wynalazków takich jak mikroprocesor PowerPC czy głowice odporne na zakłócenia magnetyczne przynosi zyski, które między innymi pokrywają koszty badań rozwojowych. Firma IBM nieustannie pomnaża swe zyski. Obrót IBM za rok 1997 wyniósł 78.5 miliardów USD.

IBM w dużym stopniu koncentruje się na współpracy naukowców z klientami. Klienci zyskują bardziej bezpośredni dostęp do laboratoriów, fachowcy zaś pomagają im znaleźć potrzebne rozwiązania. Firma nadal będzie inwestować w projekty badawcze, a kluczowe technologie zostaną zastosowane w produkcji intensywniej niż dotychczas. IBM jest największym dostawcą sprzętu informatycznego na świecie i pierwszym producentem oprogramowania. 9 fabryk należących do firmy w 5 krajach europejskich wykorzystuje najbardziej zaawansowane technologie. Niektóre fabryki zostały przekształcone w ośrodki rozwoju przemysłu, wykorzystujące i sprzedające praktyczną wiedzę IBM na temat produkcji. IBM sprzedaje swoje produkty i usługi poprzez partnerów, a także katalogi i telemarketing. Jednak klienci mają bardzo indywidualne potrzeby i często pragną rozmawiać bezpośrednio z przedstawicielami firmy.

Dziedziną, w której firma rozwinęła wiele układów partnerskich jest oprogramowanie. W celu zwiększenia możliwości wyboru i skrócenia czasu oczekiwania na gotowe produkty IBM współpracuje z setkami przedsiębiorstw europejskich tworzących programy komputerowe. Posiada w Europie 8 ośrodków rozwoju oprogramowania, w których pracują tysiące fachowców.

Usługi są jednym z ważniejszych elementów działalności firmy od chwili powstania aż do dziś. IBM zawsze oferował klientom szereg usług, między innymi inżynierię systemową czy szkolenia użytkowników. W ostatnich latach dział niezwykle się rozrósł i zróżnicował, czyniąc IBM największą firmą usługową w Europie.

Korporacja IBM nie zajmuje się wyłącznie dostarczaniem najnowocześniejszych systemów informatycznych, ale zgodnie z tradycją i zasadami aktywnie uczestniczy w życiu społecznym, wspierając te przedsięwzięcia i dziedziny, które nie mogłyby się rozwijać bez pomocy sponsorów, jak ochrona środowiska, ochrona zdrowia, kultura.

IBM ma wieloletni dorobek w zakresie ochrony środowiska, przestrzega surowych zasad higieny i bezpieczeństwa, propaguje swój kodeks wszędzie tam, gdzie prowadzą go interesy. I tak np. firma poinformowała o wyłączeniu ze wszystkich swoich procesów produkcyjnych substancji CFC (chloro- i fluoropodobnych węglowodorów), powodujących zmniejszenie ilości ozonu w atmosferze. Realizacja tego projektu była efektem wysiłku 40 ośrodków na całym świecie. Firma wyprzedziła o dwa lata obowiązujący wszystkich producentów ostateczny termin wycofania CFC z przemysłu.

IBM uczestniczy w europejskich przedsięwzięciach kulturalnych, między innymi sponsoruje noworoczny koncert w Filharmonii Wiedeńskiej, transmitowany dla melomanów na całym świecie. Finansował również wykonanie kopii prehistorycznych rysunków z jaskiń Lascaux i udostępnienie ich publiczności w Muzeum Grand Palais we Francji oraz podniesienie i odrestaurowanie statku "Mary Rose" w Anglii.

Na mocy porozumienia z Międzynarodowym Komitetem Olimpijskim IBM wyposażył w systemy komputerowe igrzyska zimowe w Lillehammer. Współpraca z Międzynarodowym Komitetem Olimpijskim była tak udana, że IBM powierzono dostawę sysytemów informatycznych zarówno letnich, jak i zimowych igrzysk olimpijskich, aż do 2000 roku. Po Lillehammer IBM zadbał o komputerowe zabezpieczenie na olimpiadzie w Atlancie, a w roku 1998 będzie w Nagano. Zasadniczym powodem, dla którego wybór padł na IBM jest - oprócz technologicznego zaawansowania - obecność firmy na całym świecie.

W różnych częściach globu IBM otworzył swoje ośrodki regionalne i narodowe (np. centrala IBM Europe znajduje się w Paryżu, a centrala IBM Eastern Europe w Wiedniu). Ośrodki lokalne prowadzą najczęściej obywatele danego kraju. Firma realizuje swoje projekty służąc jednocześnie społecznościom narodowym. Produkcja podlega regionalizacji,

ale nie zamyka się w granicach poszczególnych państw. Wszystkie gotowe wyroby stanowią obecnie rezultat pracy wielu ośrodków regionalnych, np. system IBM produkowany we Włoszech może mieć dysk twardy z Francji, układy logiczne z Wielkiej Brytanii, elementy pamięciowe z Niemiec. Jedną z firm lokalnych należących IBM World Trade EMEA jest IBM Polska.

IBM Polska

Na polskim rynku firma IBM pojawiła się już w 1923 r., otwierając w Warszawie swoje przedstawicielstwo. W następnych latach funkcjonowała nadal, dostosowując się do zmiennej rzeczywistoœci politycznej i gospodarczej. Działała zgodnie z prawem polskim jako:

* Polski Hollerith (1934 r.)
* Watson Business Machines (1938 r.)
* IBM ROECE Marketing (1966r.)
* IBM Biuro Techniczno-Informacyjne (1973 r.)
* Biuro IBM WT EMEA (1978 r.)

- od 1983 r. przedstawicielstwa krajowe:

EXIMPOL (1983-88)

IAIZ (1988-89)

DERNAN (1989-91)

1 października 1991 r. powstał IBM Polska (samodzielna firma krajowa IBM World Trade EMEA Corporation). W ciągu niespełna trzech lat IBM Polska otworzył swoje oddziały we Wrocławiu, Katowicach, Poznaniu i Gdyni, a po sześciu latach istnienia zespół powiększył się do 240 osób.

Od chwili powstania firma systematycznie i skutecznie wprowadza kolejno na nasz rynek najnowocześniejsze systemy informatyczne, przeznaczone dla różnorodnych odbiorców. Oferta obejmuje wszystkie systemy komputerowe produkowane przez IBM. Są wśród nich komputery osobiste: dla osób bez doświadczenia w pracy z komputerem oraz dla zaawansowanych użytkowników, serwery dla sieci i komputery przenośne. ThinkPad - komputer typu notebook stał się rewelacją od 1993 r. i stworzył standard w świecie notebooków. Komputery IBM RS/6000 należą do najbardziej bezpiecznych systemów otwartych. Z kolei kilkanaście modeli AS/400 jest przeznaczonych do obsługi firm różnej wielkości. Natomiast serwery IBM S/390 są najbezpieczniejszą i najbardziej wydajną platformą obliczeniową przeznaczoną dla klientów o najwyższych wymaganiach.

IBM Polska proponuje również oprogramowanie systemowe, komunikacyjne oraz użytkowe, obejmujące między innymi systemy dla banków i zakładów przemysłowych. Działalność ta owocuje wymiernymi wynikami finansowymi. Obroty firmy IBM Polska Sp. z o. o. za rok 1996 wyniosły 61 919 tys. zł.

Firma prowadzi działalność razem z partnerami handlowymi (agentami i remarketerami), których ogólnopolska sieć umożliwia dotarcie do każdego klienta, dzięki czemu ich liczba szybko się powiększa. Z ofert IBM korzystają instytucje administracji państwowej, banki, firmy ubezpieczeniowe, zakłady przemysłowe, szkoły i wyższe uczelnie.

Od początku istnienia IBM kładł szczególny nacisk na rozwój usług. Jako firma wszechstronna i otwarta, zapewnia kompleksową obsługę klienta, zaczynając od zaprojektowania systemu, poprzez dostarczenie sprzętu i oprogramowania, a kończąc na serwisie gwarancyjnym i pogwarancyjnym. Firma bierze odpowiedzialność za końcowy wynik działań. Świadcząc usługi na terenie Polski korzysta z krajowych poddostawców i podwykonawców. Jest to element świadomej polityki IBM Polska zmierzającej do rozwiajnia lokalnej gospodarki rynkowej.

W firmie nadal żywe jest przekonanie, że jej przyszłość zależy od indywidualnych umiejętności pracowników. IBM inwestuje w szkolenia handlowców i inżynierów w wielu dziedzinach pozornie nie mających związku z technologią komputerową, np. bankowość, transport, telekomunikacja, zarządzanie produkcją. IBM Polska przeznacza na kształcenie 1/5 świadczeń przypadających na jednego pracownika. Ponadto szkoli pracowników swoich partnerów, pomagając

w tworzeniu nowych miejsc pracy. Współpraca z licznymi partnerami handlowymi pozwala na rozszerzenie oferty o specjalizowane programy aplikacyjne.

W krótkim czasie firma stała się nieodłącznym elementem polskiego pejzażu gospodarczego. Między innymi zajmowała się wprowadzeniem systemu komputerowego na Giełdzie Papierów Wartościowych.

Wieloletnia pozycja lidera na światowym rynku informatycznym oraz bogate doświadczenie w instalowaniu systemów komputerowych sprawiły, że nazwa "IBM" jest synonimem najwyższej jakości, a firma - siłą napędową światowego przemysłu informatycznego. Celem IBM dzisiaj jest ścisła współpraca z klientami i odnoszenie sukcesów w poszukiwaniu najlepszych technologii na miarę XXI wieku.

Niewątpliwie Równoległe Serwery Przedsiębiorstw IBM S/390 (Parallel Enterprise Servers) wykonane są w najnowszej technologii. Ich zastosowanie pozwala na zmniejszenie kosztów przetwarzania danych poprzez Konsolidację Serwerów. Równocześnie można wykorzystywać aplikacje od różnych dostawców, integrując w ten sposób kolejne środowiska. Zmniejsza to koszty, przy zwiększeniu funkcjonalności w porównaiu do innych systemów. Więcej o tych zagadnieniach powiemy podczas Czternastych Jesiennych Spotkań PTI Mrągowo '98 we wtorek 17 listopada 1998. Nasza prezentacja rozpocznie się o godz. 15.15 - SERDECZNIE ZAPRASZAMY.

Andrzej Wyrzykowski

MHM Computer Polska Sp. z o.o.

Siedziba główna w Warszawie,
ul. Biała 4,
00-895 Warszawa
tel. 00 48 22 /620 95 92, 620 10 32
fax 00 48 22 / 620 43 89

Biuro we Wrocławiu
ul. Purkyniego 1,
50-155 Wrocław
tel./fax 00 48 71 /34 173 13

Biuro w Krakowie
ul. Zaczarowane Koło 5,
30-085 Kraków
tel./fax 00 48 12 /635 88 54

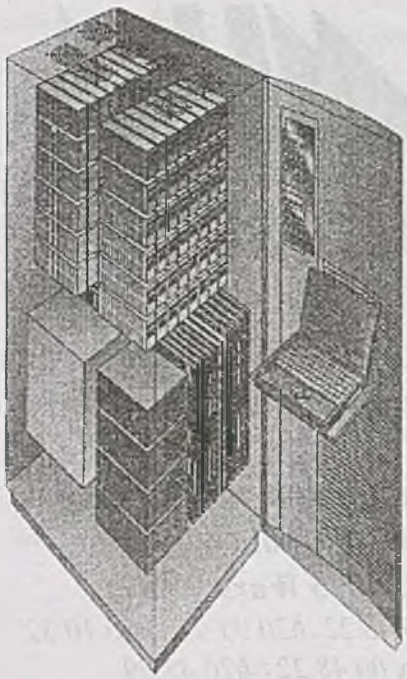
Enterprise Storage – gwarancja bezpieczeństwa danych

MHM Computer jest w Polsce jedynym, niezależnym od platformy sprzętowej dystrybutorem firmy **EMC Corporation** – producenta zewnętrznych systemów pamięciowych.

Oznacza to, że jesteśmy zdolni dostarczyć EMC Symmetrix Enterprise Storage i zainstalować do każdej platformy sprzętowej (mainframe, UNIX, AS/400, Windows NT,...). Jest to poparte wieloletnim i bogatym doświadczeniem – zainstalowaliśmy ponad 50 systemów Symmetrix (więcej niż 11 TB) na różnych platformach.

MHM Computer specjalizuje się w kompleksowym rozwiązywaniu problemów związanych przechowywaniem i zabezpieczaniem danych, backup'em, archiwizacją i disaster recovery.

Firma zajmuje się również doradztwem i projektowaniem systemów informatycznych wykorzystujących koncepcję Enterprise Storage.



Podstawową ofertą MHM Computer jest zewnętrzny system pamięciowy **Symmetrix Enterprise Storage**, charakteryzujący się:

— *najwyższym, obecnie dostępnym poziomem bezpieczeństwa,*

— *możliwością współpracy, jednocześnie z różnymi platformami sprzętowymi (mainframe ,systemy otwarte, Windows NT, AS/400 i inne),*

— *bardzo dużą skalowalnością,*

— *bogatą ofertą dodatkowych funkcji, realizowanych na poziomie systemu Symmetrix (współdzielenie plików pomiędzy różnymi systemami, możliwość tworzenia w czasie pracy dynamicznych kopii danych, oddalony mirroring i wiele innych)*

Ofertę MHM Computer w zakresie systemów pamięciowych uzupełniają:

EMC Data Manager *– scentralizowany, wydajny system backup dla całej firmy,*

EMC Cellera File Server *– niezwykle wydajny serwer plików.*

Ponadto MHM Computer jest dystrybutorem macierzy dyskowych japońskiej firmy **ADTX**, które mogą znaleźć zastosowanie u klientów o mniejszych wymaganiach związanych z pojemnością i wydajnością systemu.

Dariusz Duchowski, Krzysztof Zabielski

EMAX Sp. z o.o.
ul. Niezłomnych 1c
61-894 Poznań

Najnowsze rozwiązania w dziedzinie przetwarzania dokumentów

EMAX świadczy specjalistyczne usługi w zakresie organizacji i automatyzacji procesów zarządzania: informacją, gotówką, przedmiotem o standardowych gabarytach, zużyciem energii elektrycznej i cieplnej w obrębie budynku, a także bezpieczeństwem przeciwpożarowym i dostępem do obiektu.

Nasze działania opieramy na ścisłej współpracy z klientem. Towarzyszymy mu od momentu identyfikacji problemu przez analizę i wybór optymalnego rozwiązania, projektowanie, dostawę i wdrożenie aż po stałą opiekę serwisową nad działającym systemem.

Nasi główni klienci to największe instytucje rynku finansowego, administracja centralna, przedsiębiorstwa sektora publicznego oraz duże przedsiębiorstwa przemysłowe i usługowe. Proponowane przez nas rozwiązania z zakresu teleinformatyki i automatyki tworzymy przy współpracy z renomowanymi dostawcami najnowocześniejszych technologii.

Realizujemy rozbudowane projekty oparte na integracji systemowej i one to wyznaczają strategiczny kierunek rozwoju naszego przedsiębiorstwa.

Wprowadzenie powszechnych w państwach Unii Europejskiej, a obecnie i w Polsce, formatów dokumentów przeznaczonych do automatycznego odczytu i przetwarzania znacznie zmniejszyło ilość niezbędnych do wykonania operacji związanych z wprowadzaniem danych do systemów komputerowych, ich weryfikacją oraz archiwizacją. Oferowane przez nas systemy umożliwiają automatyczną personalizację, kodowanie, odczyt oraz archiwizację dokumentów. Obraz dokumentu, po zeskanowaniu, wykorzystywany jest do odczytania treści, która zapisana może być różnymi rodzajami pisma:

— standardowym (OCR A/B, Farrington, E13B CMC7),
— maszynowym (wszystkie czcionki),
— ręcznym (blokowym, separowanym).

Dzięki wykorzystaniu specjalizowanego oprogramowania, odczytującego OCR/ICR oraz uniwersalnej platformy programowej możliwe jest dowolne skalowanie systemów - od małych stanowisk skanująco-czytających do zintegrowanych systemów odczytu, weryfikacji i archiwizacji - przetwarzających wiele setek tysięcy dokumentów dziennie.

Systemy przetwarzają do 18000 dokumentów na godzinę. Posiadają oprogramowanie odczytujące formularze obrotu płatniczego (czeki, przelewy, wpłaty, przekazy rentowo-emerytalne), w tym przede wszystkim dokumenty w standardzie zatwierdzonym przez Związek Banków Polskich. Po odczycie i weryfikacji następuje automatyczne sortowanie dokumentów, a następnie ich archiwizacja. Uniwersalne środowisko programowe umożliwia łączenie systemów w większe, hierarchicznie zarządzane struktury.

Systemy oparte o skanery produkcyjne przetwarzają do 7000 dokumentów na

godzinę. Stosowane zazwyczaj komponenty sprzętowe i programowe zapewniają:

— skanowanie obustronne dokumentów wraz z nadrukiem adnotacji,

— filtrowanie tła i binaryzację obrazu,

— rozpoznanie treści (pismo, kody kreskowe),

— korektę i uzgodnienie odczytanych paczek dokumentów,

— konwersję danych do założonego formatu,

— transfer danych do systemów zewnętrznych,

— archiwizację obrazów.

Nasze instalacje znalazły zastosowanie w:

— bankowych stanowiskach dysponencko-kasjerskich,

— centrach przetwarzania dokumentów finansowych,

— pocztowych sortowniach przekazów rentowo-emerytalnych,

— systemach obsługi kredytów,

— centrach zarządzania dokumentacją ubezpieczeniową,

— archiwach instytucji finansowych i publicznych,

— ośrodkach informacji prasowej,

— centrach billingowych.

Dokumenty finansowe, faktury, zamówienia, umowy, wydawnictwa...

W dzisiejszym, pełnym pośpiechu świecie wciąż rośnie ilość tworzonych dokumentów i wysyłanej korespondencji.

Jej obsługa stanowi poważną pozycję w budżecie firmy, a zarządzanie ośrodkiem masowego druku i przygotowania korespondencji (np. w ramach centrum mailingowego banki, ubezpieczalnie, lub centrum bilingowego operatorzy telekomunikacyjni) jest złożonym procesem, w którym liczą się wydajność i precyzja.

Znaczenie ośrodka wysyłki korespondencji dla ogólnej efektywności przedsiębiorstwa jest często niedoceniane.

Sprawnego przygotowania korespondencji można dokonać tylko w oparciu o nowoczesne automatyczne technologie, obejmujące: transport papieru, wydruk, formatowanie dokumentów, kompletację pakietu dla klientów, kopertowanie, zaklejanie kopert, frankowanie i kontrolę poprawności procesów.

Systemy druku na żądanie zapewniają elastyczne tworzenie wydawnictw (np. instrukcje obsługi, regulaminy, materiały szkoleniowe itp.) w zależności od bieżących potrzeb.

Rozwiązujemy wszystkie problemy związane z tworzeniem dokumentów i przygotowaniem korespondencji do wysyłki począwszy od konsultacji, projektowania.

Robert Rylko

Computer Associates Sp. z o. o.
Centrum LIM
Al. Jerozolimskie 65/79
00-697 Warszawa

Zarządzanie stacjami roboczymi w ramach zintegrowanego systemu zarządzania środowiskiem IT

Zintegrowane środowisko zarządzania Unicenter TNG

Współczesne środowiska Information Technology (IT) cechuje z jednej strony heterogeniczność rozwiązań (różnorodność platform, systemów operacyjnych, protokołów, baz danych, etc.) a z drugiej strony rosnące wymagania dotyczące zakresu zarządzanych obiektów (sieci, systemy, bazy danych i aplikacje). Ponadto istotnym czynnikiem jest potrzeba skalowalności systemu zarządzania pozwalająca na rozwój zasobów IT wraz z rozwojem przedsiębiorstwa. System zarządzania Unicenter TNG jest przykładem rozwiązania które zapewnia maksymalizację we wszystkich powyższych aspektach. Jest to osiągnięte dzięki otwartej i skalowalnej architekturze oraz odpowiednim rozwiązaniom projektowym.

Od strony funkcjonalnej Unicenter TNG składa się z Framework uzupełnionego o funkcjonalność Base oraz szeregu opcji rozszerzających funkcjonalność produktu Base. W ramach funkcjonalności Base można wyróżnić:

- **Event Management**, kompleksowe zarządzanie zdarzeniami IT obejmujące zarówno monitorowanie i filtrację zdarzeń oraz automatyczne akcje w odpowiedzi na konkretne zdarzenia, jak również korelację zdarzeń,
- **Network Management,** zapewniający automatyczną wizualizację i obsługę urządzeń i elementów sieciowych,
- **Workload Management**, zarządzanie pracami, pozwalające na automatyzację prac wykonywanych okresowo i uwzględnienie czasowej hierarchi i zależności zadań oraz wykonania zadań uwzględniającego aktualne obciążenie platform w danej grupie,
- **Security Management**, zarządzanie bezpieczeństwem, zapewniające niezależną od platformy rejestrację użytkowników oraz kontrolę dostępu do definiowalnych zasobów systemu,
- **Storage Management**, zarządzanie pamięcią masową, umożliwia zaawansowane metody zabezpieczania, archiwizacji i odzyskiwania danych, zarządzanie mediami oraz hierarchiczny system plików,
- **Performance Management**, zarządzanie wydajnością systemów, dostarcza narzędzi do zbierania i analizy statystyk wydajnościowych systemu zarówno online jak i zbieranych w dłuższych okresach,
- **Problem Management**, zarządzanie problemami, obejmuje zgłaszanie, śledzenie, eskalację i rozwiązywanie problemów w środowisku IT,

- **Output Management**, zarządzanie dystrybucją raportów do końcowych użytkowników zależnie od przynależności do danej grupy.

Od strony architektonicznej Unicenter TNG składa się z czterech poziomów: Agentów, Menadżerów, Wspólnego Repozytorium Obiektowego (**CORE**) oraz **Real World Interface** – warstwy obsługi dwu lub trójwymiarowego graficznego interfejsu użytkownika. Warstwa agentów obejmuje zarówno agentów właściwych dla Unicenter TNG, realizujących poszczególne funkcje zarządcze, jak również dowolnych agentów SNMP.

Podstawowa funkcjonalność TNG jest wzbogacona o funkcjonalność opcji –dodatkowych modułów oprogramowania realizujących określone funkcje. Opcje pomocne w zarządzaniu stacjami roboczymi to przede wszystkim:

- **Asset Management Option** realizująca inwentaryzację sprzętu i oprogramowania,
- **Software Delivery Option** realizująca dystrybucję oprogramowania,
- **Remote Control Option** realizująca zdalną kontrolę nad stacją roboczą,
- **Single-Sign On Option** realizująca funkcję jednokrotnego logowania,
- **Advanced Antivirus Option** zapewniająca ochronę przed wirusami,
- **Directory Management Option** realizująca zunifikowane zarządzanie katalogami.

Zagadnienia związane z zarządzaniem stacjami roboczymi

Problemy spotykane w zarządzaniu stacjami roboczymi cechuje pewne podobieństwo do problemów zarządzania serwerami. Główne różnice wynikają ze skali, tj. liczby zarządzanych obiektów oraz, z uwagi na słabszą moc obliczeniową stacji i inny charakter procesów i zasobów, z rodzaju informacji i metod zarządzania.

Inwentaryzacja sprzętu i oprogramowania

Funkcje powyższe są realizowane przez **Asset Management Option** (AMO). Podstawowe funkcje AMO to:

- Inwentaryzacja Sprzętu (**Hardware Inventory**),
- Inwentaryzacja Oprogramowania (**Software Inventory**),
- Kontrola konfiguracji (**Configuration Control**)
- Pomiary Oprogramowania (**Software Metering**)

Inwentaryzacja hardware'u dotyczy pięciu grup informacji o systemie:

◊ *General Inventory*: typ CPU, szczegóły systemu operacyjnego, pamięć, lokalne i sieciowe sterowniki, mysz, klawiatura, karta graficzna, karta sieciowa, dane z BIOS, porty I/O etc.

◊ *Additonal Inventory*: numery seryjne twardych dysków, rozdzielczość ekranu, urządzenia z problemami (widzianymi z poziomu systemu operacyjnego), serwisy (NT) wraz z ich statusem,

◊ *DMI Inventory*: dla stacji obsługujacych DMI,

◊ *Operating System Inventory*: zasoby widziane przez system operacyjny,

◊ *Network Inventory*: szczegółowa informacja dotycząca zainstalowanych protokołów sieciowych.

Inwentaryzacja oprogramowania AMO polega na automatycznym wykrywaniu i inwentaryzacji aplikacji i oprogramowania systemowego. Oprogramowanie jest rozpoznawane z wykorzystaniem specjalnego języka skryptowego pozwalającego rozpoznawać aplikacje według różnorodnych kryteriów takich jak nazwa, rozmiar oraz data utworzenia odpowiednich plików konfiguracyjnych, wykonywalnych, dll, etc. Gotowa lista aplikacji zawiera opis około 3500 aplikacji. Typowa konfiguracja AMO powoduje konieczność wykrywania tylko podzbioru pełnej listy.

W zakres kontroli konfiguracji wchodzi zbieranie, blokowanie i monitorowanie zmian w zbiorach konfiguracyjnych zarówno systemowych jak aplikacyjnych. Dowolny zbiór ASCII może zostać zdefiniowany jako zbiór

konfiguracyjny podlegający kontroli. Ponadto język skryptowy AMO może zostać wykorzystany do odczytu, edycji i zmiany rejestrów (NT).

Pod pojęciem **Software Metering** rozumiana jest następująca funkcjonalność:

- ◊ Monitorowanie liczby użytkowników danego oprogramowania, zarówno pasywne (offline) jak i aktywne,
- ◊ Ograniczenie w używaniu danych aplikacji, jeśli został przekroczony limit dozwolony przez licencję, zarówno poprzez blokowanie jak i kolejkowanie (FIFO) użytkowników,
- ◊ Zapobieganie używaniu niektórych aplikacji (np. gier), zarówno przez specyfikację listy niepożądanych aplikacji lub listy aplikacji które mogą być używane,
- ◊ Audyt używanego oprogramowania: programów exe, dll oraz serwisów, niezależnie od oprogramowania zinwentaryzowanego,
- ◊ Zapobieganie instalacji niepożądanego oprogramowania, zarówno poprzez zablokowanie możliwości uruchamiania programów typu install.exe lub setup.exe, jak również w bardziej wyrafinowany sposób, stosując directory synchronization, wymuszając identyczną strukturę i zawartość katalogów na maszynie referencyjnej i dowolnej stacji roboczej.

Dystrybucja oprogramowania

Software Delivery Option (SDO) zapewnia centralnie zarządzaną instalację, konfigurację, dystrybucję, aktywację, weryfikację, aktualizację i możliwość deinstalacji oprogramowania. Dystrybucja oprogramowania składa się z trzech etapów:

- *rejestracji oprogramowania,* czyli zdefiniowania danego pakietu w ramach SDO,
- *zaplanowania dystrybucji,* czyli przesłania pakietu zależnie od zastosowanej architektury dystrybucji,
- *monitorowania dystrybucji.*

Oprócz prostej dystrybucji oprogramowania w sieci lokalnej, architektura SDO umożliwia wielopoziomową dystrybucję poprzez sieci rozległe. Podstawowe elementy architektury SDO to:

- **Enterprise serwer,** umieszczony w centralnym punkcie zarządzania,
- **Fanout serwer,** służy do redukcji obciążenia sieci przy dystrybucji oprogramowania w sieciach rozległych,
- **Local serwer,** sterujący dystrybucją w wydzielonych obszarach sieci,
- **Staging serwer,** spełniający podobną rolę jak fanout serwer w sieci lokalnej,
- **Target computers**, stacje końcowe z zainstalowanym agentem SDO.

Zdalna kontrola

Funkcjonalność **Remote Control Option** pozwala na zdalne przejęcie kontroli nad zdalną stacją roboczą wraz z pełnym przejęciem ekranu. Może się to odbywać w kontekście:

- Obsługi przez personel Help Desk pozwalającej na bezpośredni wgląd i manipulację zdalną stacją roboczą,
- Zdalnej pracy z domu,
- Edukacji i treningu.

Zarządzanie katalogami

Directory Management Option zapewnia scentralizowne zarządzanie katalogami, m.in. pozwala na audyt katalogów i dokonywanie zmian offline bez interwferencji ze środowiskiem użytkownika w czasie godzin pracy.

Ochrona i bezpieczeństwo

Single Sign-On Option umożliwia automatyczne rejestrowanie się do aplikacji oparte na jednokrotnej rejestracji do systemu, eliminuje więc potrzebę pamiętania i zapisywania wielu identyfikatorów użytkownika i haseł dla każdej używanej aplikacji.

Advanced Antivirus Option jest zintegrowanym rozwiązaniem antywirusowym

zapewniającym m.in. centralną konsolę i logi administracji antywirusowej, skanowanie i leczenie zbiorów w czasie rzeczywistym, automatyczne uaktualnianie listy wykrywanych wirusów.

Integracja *Desktop Management* z systemem kompleksowego zarządzania IT

Integracja może być rozpatrywana w kontekście integracji produktów do zarządzania stacjami roboczymi z produktem zarządzania całym środowiskiem IT, jak również integracji - współpracy produktów do zarządzania desktopami pomiędzy nimi. Unicenter TNG wraz z opcjami zapewnia integrację w obu powyższych aspektach. Integracja systemów do zarządzania desktopami z systemem zarządzania zasobami IT odbywa się na kilku płaszczyznach:

- Poziom **zdarzeń** w środowisku IT, obsługiwanych zarówno przez przekazywanie komunikatów oraz wspólny dostęp do logów i automatyczne reakcje na zdarzenia obejmujące obszar funkcjonalności różnych modułów,
- Poziom **ochrony**, umożliwiający jednolite podejście do problemu bezpieczeństwa,
- Poziom **prac okresowych i kalendarzy**, pozwalający na koordynację automatycznych prac administracyjnych w oparciu o wspólne kalendarze,
- Poziom **Help Desk**, pozwalający na wymianę informacji o środowisku użytkownika, automatyczne śledzenie i generowanie kart problemów,
- Poziom **interfejsu graficznego**, pozwalający wizualizować środowisko wraz ze stanem procesów zarządczych,
- Poziom **wspólnego repozytorium** umożliwiający współdzielenie wybranej informacji.

Podsumowanie

Unicenter TNG wraz z opcjami **Asset Management Option, Software Delivery Option, Remote Control Option, Directory Management Option, Single Sign-On Option** oraz **Advanced Antivirus Option** stanowi wszechstronne i zintegrowane rozwiązanie do zarządzania zarówno całym środowiskiem IT jak również stacjami roboczymi.

Ryszard Krawczyński

ORACLE Polska Sp. z o.o.
Sienna Center, ul. Sienna 75, 00-833 Warszawa

ORACLE Applications

Oracle Applications - Planowanie Łańcucha Dostaw *)

WSTĘP

Systemy Planowania i Optymalizacji Łańcucha Dostaw są logicznym etapem ewolucyjnym systemów planistycznych. Ewolucja ta opiera się na założeniach "myślenia systemowego", czyli traktowania systemu jako nierozerwalnej całości. Jednym z podstawowych założeń tej teorii jest fakt, iż optymalizacja indywidualnego elementu systemu może prowadzić do suboptymalizacji całego procesu. Na takim sposobie myślenia opierają się też sprawdzone techniki jak Całościowe Zarządzanie Jakością, Reengineering, Produkcja Potokowa i Teoria Ograniczeń. W książce [2] Dr. Eliyahu Goldratt twierdzi, iż "suma lokalnych optimum nie musi być równa globalnemu optimum" Jedynie poprzez obserwację, zrozumienie i ulepszanie systemu jako nierozerwalnej całości możemy doprowadzić do jego skutecznej optymalizacji. To założenie ma szczególną wagę w procesie optymalizacji łańcucha dostaw.

Niniejszy dokument rozpoczniemy od definicji terminu "łańcuch dostaw". Potem wytłumaczymy, czemu Optymalizacja Łańcucha Dostaw jest tak ważnym problemem. W następnej części opowiemy o ważnej roli systemów informatycznych w zarządzaniu łańcuchem dostaw oraz o wadze elementów planowania w optymalizacji łańcucha dostaw. Dalej przedstawimy historię i ewolucję systemów planistycznych oraz zmiany w otoczeniu i narastającą konkurencję które wpłynęły na szybki rozwój systemów planistycznych. Z tego punktu przejdziemy do dyskusji o aktualnych czynnikach konkurencyjnych.

ŁAŃCUCH DOSTAW

Łańcuch dostaw to sieć organizacji przetwarzających surowiec w dobra konsumpcyjne. Jedna ze stosowanych ostatnio definicji brzmi: "łańcuch dostaw to system, z pomocą którego organizacje dostarczają swoje produkty i usługi do klientów (...), ich wspólnym celem jest stosowanie jak najlepszych metod zrealizowania tej dostawy" (Poirer, [6]). Sukces całego łańcucha jest zależny od tego, czy wymagania końcowego klienta są spełnione, czy nie. To jest właśnie główny cel poszczególnych organizacji uczestniczących w łańcuchu dostaw.

Korzyści i zyski dla indywidualnej organizacji płynące bezpośrednio z łańcucha dostaw muszą stanowić cel podrzędny, w stosunku do korzyści całościowych, obejmujących własne interesy. Kapitalizm i praktycznie całe społeczeństwo funkcjonuje właśnie w oparciu o zdrowy rozsądek i dbanie o własne interesy. Rynek wymusza na

*) Opracowano na podstawie artykułu Scotta, M. „Supply Chain Planning and Optimization" Oracle Corporation, April 1998

potencjalnych konkurentach współpracę, nagradzając ich za udane wspólne inicjatywy. Klient może kupić zepsuty produkt, ale tylko raz. Jeśli zaś sprzeda mu się produkt odpowiedniej jakości, nawet wyższym kosztem, wytwarza się powtarzalność transakcji oraz dobre referencje. Tak właśnie myślą dostawcy produktów do końcowego klienta, nie dlatego, że z dobrego serca myślą o interesach klienta, lecz dlatego, że jest to dla nich na dłuższą metę bardziej opłacalne.

Jeśli koszty są przenoszone z jednego ogniwa łańcucha dostaw na inny poprzez negocjacje, pozostaną one nadal w łańcuchu i nadal wpłyną na ceny płacone przez końcowego klienta. Stanowi to także ograniczenie zysków do późniejszego podzielenia. Koszty przeniesione na inne ogniwo łańcucha mają także tendencję do powracania. Może to się zdarzyć, kiedy np. brak zapasów spowoduje ograniczenie siły negocjacji, lub przy zredukowanym poziomie jakości produktu czy usług. Przypomina to łapanie kreta – likwidacja norki nie spowoduje niczego innego jak powstanie kilku innych w okolicy, w najmniej oczekiwanych miejscach. Trzeba zlikwidować samo źródło problemu, czyli zamiast niszczyć norki - złapać kreta. Koszty które firma próbuje zrzucić z własnych barków poprzez negocjacje z partnerami, zamiast je eliminować u źródła, będą miały tendencję do "wyskakiwania" w najmniej oczekiwanym momencie.

Aby prawidłowo podejść do zadania optymalizacji łańcucha dostaw, koszty trzeba eliminować z łańcucha, a nie rozkładać na jego członków. Wymaga to jednak wyższego stopnia współpracy między organizacjami wchodzącymi w skład łańcucha. Członkowie muszą zdawać sobie sprawę z obustronnych korzyści, jakie mogą uzyskać poprzez optymalizację łańcucha dostaw. W książce Brandenburger i Nalebuff [1] stwierdziają, że: "biznes jest współpracą przy pieczeniu tortu i konkurencją przy jego dzieleniu". Optymalizacja łańcucha dostaw jest zależna od tego, czy utrafimy w punkt równowagi między elementami konkurencji a współpracy.

Jednym z podstawowych problemów w procesie optymalizacji systemu jest określenie zakresu problemu który mamy rozwiązać. Przy definiowaniu tego problemu mamy dwa cele, które zaprzeczają sobie nawzajem. Z jednej strony, jeśli zdefiniujemy problem zbyt wąsko, możemy spowodować suboptymalizację, nie obejmując całego systemu. Z drugiej strony, jeśli zdefiniujemy problem zbyt szeroko, nie będziemy mogli go uchwycić. Mamy tu więc do czynienia z równowagą pomiędzy "umiejętnością spojrzenia na całość sprawy" i "braniem na barki więcej, niż zdołamy unieść". Mamy jednak pomoc w postaci technologii, która umożliwiła poszerzenie zakresu problemu, eliminując częściowo problem niemożności uchwycenia całości. Technologia ta to informatyka.

TECHNOLOGIA INFORMATYCZNA

Zanim wprowadzono do powszechnego użycia systemy komputerowe, w wielu przypadkach niemożliwe było spojrzenie na kilka problemów planistycznych jako na jeden, nierozerwalny system. Postępy w technologiach komputerowych i telekomunikacyjnych umożliwiły ewolucję systemów planistycznych tak, aby obejmowały większe zakresy problemowe. Ten sam trend z resztą dał się odczuć w innych naukach. To właśnie możliwość zarządzania większymi, bardziej złożonymi modelami doprowadziła do znacznych postępów w dziedzinach jak fizyka, chemia czy nauki społeczne.

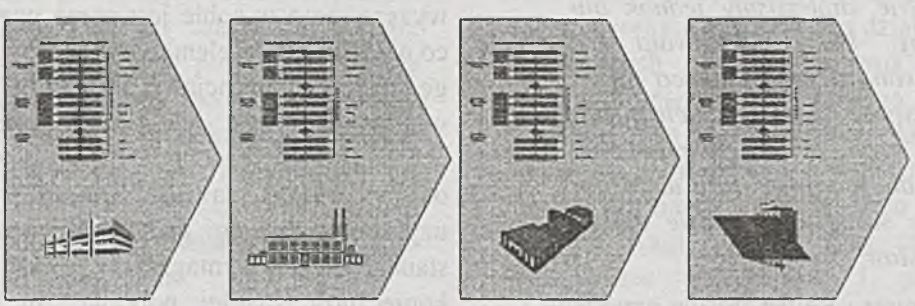
Postępy technologiczne umożliwiły także radykalną poprawę w zakresie procesów przedsiębiorstwa. Jako przykład spójrzmy na postępy w zakresie telekomunikacji takie jak elektroniczna wymiana danych (EDI, Electronic Data Interchange) czy elektroniczny przelew funduszy (EFT, Electronic Funds Transfer). Technologie te umożliwiły dokonywanie zakupów i realizację opłat w sytuacjach, gdzie tradycyjne metody jak telefon i poczta okazywały się bezużyteczne. Postępy technologiczne wciąż otwierają nam nowe drzwi. Jak napisał Poirer, "technologia informatyczna stała się czymś więcej, niż pupilkiem firm pragnących ulepszeń organizacyjnych; stała się ona koniecznością w optymalizacji łańcucha dostaw" [6].

HISTORIA SYSTEMÓW PLANISTYCZNYCH

Wczesne systemy planistyczne traktowały każdy element jako odrębny. We wczesnych latach 60. Nowe koncepcje wraz z nowymi technologiami dały zaczątek nowoczesnym produkcyjnym systemom planistycznym. Połączenie koncepcji obliczania popytu na składniki z technologią komputerową dało nową technikę zwaną Planowaniem Potrzeb Materiałowych, czyli MRP (Material Requirements Planning). MRP stało się znaczącym przełomem i fundamentalnie zmieniło spojrzenie na problemy planistyczne.

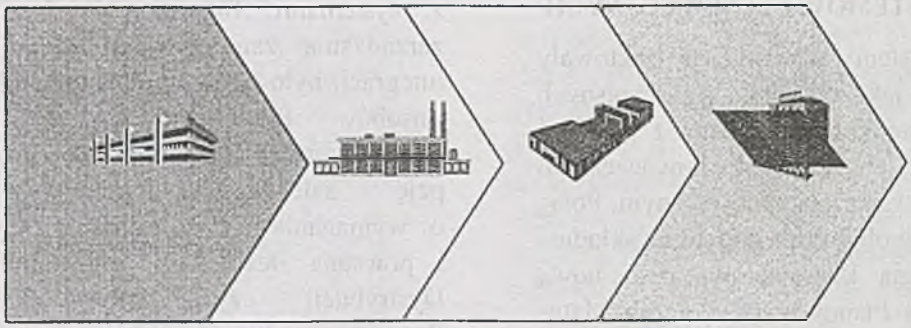
Podstawowa koncepcja MRP rozwijała się z czasem. Najpierw poszerzono koncepcję niezależnego popytu, odkrywając, że wymagania w zakresie robocizny także mogą być obliczane i wyrażane w relacji do planowanej produkcji. Powstała technika Planowania Zapotrzebowania na Zdolności Produkcyjne, czyli CRP (Capacity Requirements Planning). Proces planistyczny rozszerzono na wiele stopni szczegółowości, tworząc Planowanie Spływu Produkcji (MPS, Master Production Scheduling) oraz Przybliżone Planowanie Zdolności Produkcyjnych (RCCP, Rough Cut Capacity Planning). Systemy planistyczne zintegrowano z innymi systemami, w tym z systemami finansowymi i systemami zarządzania zamówieniami. Wynikiem tej integracji było powstanie techniki Planowania Zasobów Produkcyjnych czyli MRPII (Manufacturing Resource Planning). Koncepcję zależnego popytu poszerzono o wymagania w zakresie dystrybucji i powstała technika Planowania Potrzeb Dystrybucji czyli DRP (Distribution Requirements Planning).

W latach 80-ych do procesów produkcyjnych zaczęto podchodzić wedle nieco innej filozofii, która zmieniła znacząco spojrzenie i sposób korzystania z systemów planistycznych. Powstała technika Produkcji Na Czas (JIT - Just In Time Manufacturing), która z czasem rozwinęła się w Produkcję Potokową (Flow Manufacturing). Wprowadzono koncepcję systemów ciągnących (pull systems) oraz eliminację działań nie dodających wartości. Zarządzanie Jakością (TQM, Total Quality Management) wprowadziło koncepcję myślenia systemowego oraz statystyczne metody rozwiązywania problemów. Teoria Ograniczeń (TOC, Theory of Constraints) dodatkowo umocniła założenie, iż optymalizacja podprocesu prowadzi zazwyczaj do suboptymalizacji całości.



MRPII i DRP: Wiele oddzielnych iteracyjnych kroków

Na początku lat 90. zwiększona złożoność organizacji produkcyjnych, zwiększające się naciski ze strony konkurencji oraz postępy w technologii informatycznej doprowadziły do dalszych zmian w systemach planistycznych. Zostały one rozszerzone tak, aby obejmować kilka zakładów produkcyjnych jednocześnie. Pierwsza wersja tej koncepcji znana była pod nazwą Planowania Zasobów Przedsiębiorstwa (ERP, Enterprise Resource Planning). Model ten poszerzono tak, aby objął także dostawców i klientów. W obecnej chwili model ten znany jest jako Planowanie Łańcucha Dostaw (SCP, Supply Chain Planning).



Planowanie Łańcucha Dostaw: jeden etap planistyczny

Tak więc następny etap ewolucyjny modeli planistycznych i obecny stan rzeczy w tej dziedzinie to właśnie Planowanie Łańcucha Dostaw (Supply Chain Planning - SCP). Koncepcja SCP znana jest już od lat, lecz dopiero w chwili obecnej firmy rozpoczynają wdrażanie jej założeń w życie. W wydanej w roku 1996 książce Poirer i Reiter [6] opisują swoje badania przeprowadzone na ponad stu przedsiębiorstwach:

"Przeprowadzając badania, stwierdziliśmy, iż praktycznie każda firma z jaką się kontaktowaliśmy miała takie same założenia misji - ciągła poprawa, globalna jakość, najkrótszy cykl produkcyjny, najmniejszy poziom zapasów i najwyższy poziom satysfakcji końcowego klienta, przy jednoczesnym maksymalizowaniu dochodów. Nie znaleźliśmy jednak ani jednej firmy, która zbudowała już zoptymalizowaną sieć obejmującą całość przedsiębiorstwa. Mimo wielkich nakładów pracy tych przedsiębiorstw, nie widzieliśmy ani jednego, które uzyskałoby widoczny sukces w zakresie całego łańcucha dostaw".

Następnie Poirer i Reiter opisują przykłady przedsiębiorstw jak Walmart, Dell Computer, Rubbermaid, Procter & Gamble, K-mart i innych firm, które osiągają bardzo dobre wyniki dzięki postępującemu wdrażaniu technik optymalizacji i planowania łańcucha dostaw.

Planowanie Łańcucha Dostaw rozszerza model planistyczny, opierając się na rozwinięciu znanych koncepcji do następnego logicznego poziomu. Na przykład koncepcję zależnego popytu rozwinięto dowodząc, iż popyt jednego ogniwa łańcucha jest zależny od popytu w ogniwach znajdujących się w dole łańcucha dostaw. Techniki systemu typu ciągnącego (pull) rozszerzono tak, aby sięgały aż do wymagań końcowego klienta. Analizę wartości dodanej i eliminację odpadów rozszerzono o techniki wyszukiwania i eliminowania odpadów w związkach i transakcjach występujących między poszczególnymi ogniwami łańcucha.

W następnym rozdziale przedstawiono czynniki, które doprowadziły do konieczności powstania techniki Planowania Łańcucha Dostaw.

CZYNNIKI KONKURENCYJNE

Sytuacja na konkurencyjnym rynku, z jaką spotykają się dzisiejsze firmy, wciąż się zmienia. Tempo zmiany oczekiwań na coraz wyższe samo w sobie jest coraz większe. To, co niedawno było elementem dającym przewagę nad konkurencją dziś jest warunkiem wejścia na rynek. Jakość, szybkość, elastyczność, prędkość odpowiedzi na wymagania; obszary, które dawały firmom możliwość uzyskania przewagi nad konkurencją są dziś standardowym wymaganiem klienta. Istnieje konieczność ciągłej poprawy, by pozostać konkurencyjnym, a szczególnie jeśli chce się uzyskać przewagę na rynku. Poniżej przedstawiamy niektóre tylko z wyzwań, z jakimi na co dzień spotykają się dzisiejsze przedsiębiorstwa.

Globalizacja

Trend do konkurowania na rynku globalnym w ostatnim czasie stał się szczególnie widoczny. Postępy technologiczne takie jak Internet pozwalają nawet małym firmom zaistnieć na rynku globalnym. Jedną z konse-

kwencji takiego stanu rzeczy jest fakt, że końcowy odbiorca, konsument, ma znacznie większą możliwość wyboru. Oznacza to jednak także, że standardy, którym muszą sprostać wszystkie firmy to standardy globalne, nie zaś, jak dotychczas, lokalne.

Przeniesienie mocy decyzyjnej w stronę konsumenta i handlu detalicznego

Zwiększające się oczekiwania konsumentów i poszerzająca się gama alternatyw przeniosły siłę negocjacyjną w ręce klienteli oraz przedstawicieli handlu detalicznego. Siła ta wykorzystywana jest często przez sprzedawców detalicznych do wymuszania obniżek cenowych na dostawcach. Odpowiedzialność za elementy logistyczne także została przepchnięta w górę łańcucha. Wynegocjowane obniżki cenowe są przenoszone często bezpośrednio na klientów (strategie "codziennie niskich cen"). Obniżki cenowe, bez eliminacji kosztów z łańcucha dostaw, prowadzą do ograniczenia marży u producentów i dostawców w górnej części łańcucha.

Coraz krótszy cykl życia produktu

Efektywny i zyskowny cykl życia produktów musi być coraz krótszy. Zmiany technologiczne postępują w coraz większym tempie. W tym samym czasie technologie komunikacyjne jak WWW powodują redukcję bariery wejścia nowego produktu na rynek. Czas dostawy nowego produktu na rynek oraz szybkość odpowiedzi na zmieniające się wymagania klientów stają się coraz ważniejszymi elementami utrzymania odpowiedniego poziomu konkurencyjności.

Masowe dopasowywanie

Oczekiwania klientów stają się coraz większe. Dawniej oczywistym założeniem był fakt, iż standardowy produkt był dostarczany szybko, zaś produkt na specjalne zamówienie docierał do klienta nieco dłużej. To zjawisko traktowano jako sensowny kompromis między stopniem dopasowania i ilością czasu jaką klient czekał na dostawę. Wiele wczesnych technik JIT koncentrowało się na standaryzacji produktów tak, aby wyrównać popyt i osiągnąć efektywność poprzez skalę dostawy. Była to kontynuacja filozofii Henry Forda: "samochód może być w dowolnym kolorze, pod warunkiem, że jest czarny". Wczesne oferty firm japońskich wchodzących np. na rynek samochodowy w USA opierały się właśnie na tej strategii.

Czas zmienił znacznie tę sytuację. Klienci nie chcą już zaakceptować kompromisu między czasem dostawy a stopniem dopasowania. Obecnie oczekują oni dopasowanych produktów, dostarczanych szybko i po niskiej cenie. Tak więc konieczne stało się produkowanie dopasowanych do wymagań klienta produktów szybko, tanio i przy zachowaniu wysokiej jakości.

Rdzeń kompetencji i zlecanie na zewnątrz

Trend pionowej integracji uległ odwróceniu. Downsizing spowodował, iż wiele organizacji ma obecnie ograniczane zasoby. Aby utrzymać się na rynku, firmy muszą skoncentrować się na tym, co robią najlepiej i na obszarach, gdzie dodają najwięcej wartości do produktu czy usługi. Wykształcił się termin "rdzenia kompetencji" określający właśnie takie obszary organizacji. Wszelkie działania "nie-rdzeniowe" mogą być zlecone na zewnątrz. Konsekwencja tego zjawiska jest taka, iż coraz większa liczba działań i zasobów koniecznych do przetworzenia surowca na produkt konsumencki jest poza bezpośrednią kontrolą firmy. Prowadzi to do zwiększających się poziomów współzależności między partnerami handlowymi, a także do zwiększania się wagi zewnętrznych czynników na wewnętrzne planowanie i podejmowanie decyzji.

Dynamiczne, heterogeniczne przedsiębiorstwa i łańcuchy dostaw

Systemy MRP i MRPII zaprojektowano z myślą o modelowaniu i planowaniu działań indywidualnych zakładów. Modele w tych systemach reprezentowały więc relatywnie statyczne i homogeniczne środowiska. Pojedynczy zakład podejmuje decyzje dotyczące podstawowej metodologii produkcyjnej, procesów przedsiębiorstwa, standardów technologicznych, itp. centralnie. Technika

ERP, do pewnego stopnia, na początku także dotyczyła takiego samego modelu. Model ERP służy bowiem do planowania wszystkich elementów pojedynczej organizacji, tak więc istnieją tu jeszcze pewne założenia standaryzacji procesów i procedur.

Istniejące systemy modelowania muszą potrafić dać sobie radę z dynamicznymi i eterogenicznymi przedsiębiorstwami i łańcuchami dostaw. Wykupywanie i reorganizacje mogą drastycznie zmienić model i wprowadzić nowe elementy lub technologie nawet wewnątrz przedsiębiorstwa. Nie jest więc bezpieczne podejmowanie jakichkolwiek założeń standaryzacyjnych i dotyczy to całej długości łańcucha.

Jak *ORACLE APPLICATIONS* pozwalają na optymalizację łańcucha dostaw

Globalna wizualizacja

Oracle Applications zapewniają widoczność konieczną do zarządzania popytem i podażą na wszystkie zasoby w łańcuchu dostaw. Zarządzanie i optymalizacja łańcucha dostaw wymaga dostępu do wszystkich możliwych informacji o strukturze i sposobie działania łańcucha. Interfejs użytkownika Pulpitu Planisty (Planner Workbench) intuicyjnie, elastycznie i w prosty sposób prezentuje wszystkie informacje dotyczące dostaw, popytu, a także informacje o stanach wyjątkowych i wszelkie inne dane konieczne do zarządzania łańcuchem dostaw.

Modelowanie Łańcucha Dostaw

Oracle Applications pozwala zdefiniować i modelować cały łańcuch dostaw. Planowanie i optymalizacja łańcucha dostaw wymaga możliwości modelowania i zarządzania dużym, złożonym i heterogenicznym procesem. ***Oracle Applications*** pozwala na zdefiniowanie wszystkich organizacji biorących udział w łańcuchu dostaw, w tym także klientów i dostawców. Można zdefiniować związki między organizacjami i modelować przepływ produktów oraz materiałów przez poszczególne organizacje. Można modelować także wszelkie transakcje między firmami konieczne do zarządzania finansowym przepływem produktów. Oprogramowanie pozwala także zdefiniować standardowe metody dostawy i czasy tranzytowe.

Z pomocą zasad pozyskiwania materiałów źródłowych i Struktur Dystrybucyjnych (BOD, Bills Of Distribution) można zdefiniować dokładny przepływ produktu przez łańcuch dostaw. Zasady pozyskiwania materiałów oraz struktury dystrybucyjne mogą być przypisywane do różnych poziomów przepływu. Mogą być one przypisane globalnie, do danej organizacji, lub też do zdefiniowanej kategorii elementów, czy wreszcie do elementu lub kategorii jedynie w zakresie jednej organizacji. Przypisania są hierarchiczne, a zasady szczegółowe są nadrzędne dla zasad ogólnych. Pozwala to na szybkie i proste zdefiniowanie sieci łańcucha dostaw z pomocą zasad standardowych, a następnie zastąpienie ich szczegółowymi dla sytuacji wyjątkowych.

Przypisania mogą być dzielone, co pozwala na modelowanie złożonych związków, np. produktów rozłożonych na produkcję lokalną oraz zlecenia zewnętrzne, czy też produktów wytwarzanych etapowo w różnych zakładach. Przypisania mogą także dotyczyć określonych czasookresów, co pozwala na modelowanie warunków zmieniających się w czasie. Na przykład produkty które w przyszłości mają być zlecane na zewnątrz mogą być modelowane właśnie poprzez uzależnienie procesu pozyskiwania materiałów od daty.

Przypisania można grupować w zestawy. Umożliwia to tworzenie różnych przypisań dla celów planowania i dla z góry obiecanych zamówień. Pozwala to także na definiowanie wielokrotnych zestawów przypisań do symulacji wielokrotnych scenariuszy. Daje to silne możliwości symulacji i możliwość oceny alternatyw strategicznych.

Konsolidacja planowania

Oracle Applications korzysta z zaawansowanej architektury planowania skonsolidowanego. ***Oracle Applications*** posiada także zaawansowany, oparty na operacjach pamięciowych, generator planistyczny umożliwiający bardzo szybkie tworzenie zadań planistycznych.

Planowanie wielozakładowe

Oracle Applications pozwala zachować elastyczność planowania, umożliwiając planowanie w zakresie jednej organizacji, dowolnego podzbioru organizacji lub wszystkich organizacji jednocześnie. Planowanie wielozakładowe pozwala także na zrównoważenie wszystkich elementów dostawy oraz popytu w każdej organizacji, a także całości interakcji między organizacjami. Wszystko dzieje się w zakresie jednego, skonsolidowanego zadania planistycznego. Możliwe jest też planowanie interakcji z wcześniej zdefiniowanymi klientami i dostawcami. Cały łańcuch dostawy może być rozplanowany w zakresie jednego, szybkiego, przeprowadzonego w pamięci komputera zadania planistycznego.

Jednoczesne planowanie w zakresie wszystkich zasobów

Oracle Applications pozwala na symultaniczne planowanie wykorzystania wszystkich zasobów w łańcuchu dostaw. Z pomocą kwitów materiałowych (oraz możliwości ich projektowania), trasowania, kwitów na zasoby, zasad pozyskiwania materiałów oraz kwitów dystrybucyjnych można zdefiniować wszelkie konieczne związki.

Zaawansowany system planistyczny ***Oracle Applications*** pozwala na połączenie funkcji planowania produkcji, wymagań w zakresie zasobów, nadrzędnego planowania harmonogramów produkcji, przybliżonego planowania mocy produkcyjnej, planowania wymagań w zakresie materiałów, planowania wymagań w zakresie mocy produkcyjnych, szczegółowego planowania harmonogramów oraz planowania wymagań w zakresie dystrybucji. Wszystkie zasoby, na wszystkich poziomach, w każdej organizacji, mogą być rozplanowane szybko w jednym, przeprowadzonym w pamięci komputera, zadaniu planistycznym.

Interaktywne planowanie on-line

Oracle Applications pozwala także na interaktywne, zmienne planowanie on-line (zadania realizowane są także w pamięci komputera, przez co przetwarzanie zajmuje mało czasu). Zmiany organizacyjne pojawiają się w firmach wszędzie i wciąż, więc planiści potrzebują narzędzia, które pomoże im ocenić wpływ nowych informacji oraz pozwoli na szybkie podejmowanie decyzji i szybkie zmiany. ***Oracle Applications*** pozwala na szybką symulację zmian popytu, sprzedaży czy dostępności zasobów. Zmiany można wprowadzać do aktywnego planu, można także tworzyć nieograniczoną liczbę alternatywnych scenariuszy.

Skonsolidowane zarządzanie przez wyjątki

Oracle Applications dostarcza także narzędzi umożliwiających zarządzanie przez wyjątki. Sam zakres informacji w systemie planowania łańcucha może przytłoczyć planistę, nic więc dziwnego, że branie wszystkiego na raz pod uwagę jest praktycznie niemożliwe. Dostęp do danych nie wystarczy. Konieczny jest systematyczny proces przypisywania priorytetów i filtrowania informacji.

Informacje o wyjątkach prowadzą osoby podejmujące decyzje bezpośrednio do zdarzeń, gdzie konieczne jest podjęcie natychmiastowych działań. Filtry oraz elastyczne kryteria wyszukiwania pozwalają planistom na wybranie kilku przypadków krytycznych z ogromu trywialnych. Wszystkie sytuacje wyjątkowe, które mogą zagrozić sprawnej operacji łańcucha dostaw mogą być rozwiązane z jednego punktu wejścia. Przepływ zadań oraz alarmy pozwalają na automatyzację powiadamiania a także działań korekcyjnych. Razem, te nowe narzędzia pozwalają decydentom na szybką ocenę stanu łańcucha dostaw, identyfikację koniecznych działań korekcyjnych oraz szybkie dokonanie korekty i rozwiązanie problemu.

Zautomatyzowane podejmowanie decyzji

Oracle Applications dostarczają szerokiego zakresu narzędzi do automatycznego, rutynowego podejmowania decyzji. Jeśli tylko dana decyzja może być podjęta na podstawie przesłanek, jeśli kryteria można skwantyfikować i zakodować, to taką decyzję można zautomatyzować. ***Oracle Applications*** pozwalają na zautomatyzowanie szerokiego zakresu decyzji. Automatyzacja może dotyczyć zarówno prostych, powtarzających się zadań, jak i zadań bardziej złożonych, opartych na

skomplikowanych zasadach podejmowania decyzji (np. przepływ zadań). Pozwala to na optymalizację i planowanie w oparciu o ograniczenia.

Automatyzacja rutynowych, powtarzających się sytuacji pozwala planistom i decydentom na koncentrację na krytycznych, wyjątkowych sytuacjach o wysokim priorytecie, nie zajmując ich myśli drobiazgami w postaci zadań powtarzalnych. Wśród przykładów znajdziemy np. automatyczne wydawanie planowych zamówień spełniających przez użytkowników kryteria, czy też automatyzację procesów zapłaty za fakturę, ale także automatyczne dostrajanie procesów związanych z pasywami i aktywami.

Systemy zdolne do przetwarzania przepływu zleceń mogą pozwolić na automatyzację jeszcze bardziej złożonych zadań. Zasady przepływu zadań (workflow) mogą odtworzyć np. przepływ informacji w procesie akceptacyjnym. Możliwe jest automatyczne "przekazywanie" elektronicznych dokumentów z biurka do biurka do akceptacji poszczególnych poziomów kierowniczych. Systemy zarządzające przepływem zadań mogą także automatycznie wytyczać trasę przepływu informacji w oparciu o sytuacje wyjątkowe.

Planowanie w oparciu o ograniczone zasoby

Oracle Applications dostarcza pełnego zakresu zaawansowanych narzędzi planistycznych pozwalających na planowanie w oparciu o ograniczone zasoby. Systemy planowania z ograniczonymi zasobami pozwalają na dalsze redukcje cyklu planowania. Systemy planowania w oparciu o ograniczone zdolności pozwalają na modelowanie kryteriów decyzyjnych wykorzystywanych przez planistów do podejmowania decyzji. Pozwala to na dalszą redukcję cyklu planowania poprzez redukcję liczby iteracji oraz czasu poświęconego na utrzymanie planu.

Z pomocą ***Oracle RHYTHM*** można jednocześnie brać pod uwagę wszystkie ograniczenia - materiały, moce produkcyjne, operatorów, narzędzia, itp., a następnie wytworzyć w jednym kroku realistyczny plan operacyjny. Cały plan jest tworzony wyjątkowo szybko, ponieważ RHYTHMS także dokonuje wszystkich obliczeń w pamięci komputera.

Elastyczna Architektura

Łańcuch dostaw to otoczenie bardzo dynamiczne. Procesy przedsiębiorstwa dziś traktowane jako wewnętrzne jutro mogą być zlecone innej firmie. Zakupy i ich różnorodność mogą spowodować zmiany w przedsiębiorstwie pociągające za sobą zmiany w przepływie materiałów, finansów czy informacji. ***Oracle Applications*** to elastyczna architektura, która pozwala dostosowywać się do zmieniających się warunków.

Zasady pozyskiwania materiałów źródłowych oraz przypisania pozwalają na modelowanie warunków zmiennych w czasie. Szeroki zakres opcji profilowych pozwala na skonfigurowanie aplikacji tak, aby odzwierciedlała specyficzne procesy przedsiębiorstwa. Pozwalają one także na szybką rekonfigurację aplikacji kiedy zmieniają się warunki lub metody pracy. Pozwala to firmom sprostać dynamicznemu otoczeniu i ciągle ulepszać procesy, nie będąc krępowanym ograniczeniami systemowymi.

Technologia przepływu zadań pozwala organizacjom na kodyfikację wewnętrznych procesów przedsiębiorstwa i dopasowanie systemu do specyficznych procedur wewnętrznych. Jako przykład, zamówienia na zakup materiałów mogą być przekazywane do akceptacji i elektronicznie przekazywane dostawcom, w zależności od zasad zdefiniowanych w modelu przepływu zadań.

Kreator Implementacji (Implementation Wizard) w ***Oracle Applications*** poprowadzi użytkowników "za rączkę" przez proces konfiguracji aplikacji. To silne narzędzie pomaga przyspieszyć proces implementacji i zmiany wersji oprogramowania, konfigurując aplikacje w oparciu o specyficzne dla organizacji procesy przedsiębiorstwa.

Te i inne narzędzia powodują, iż ***Oracle Applications*** stanowi silne i elastyczne narzędzie do zarządzania dynamicznym łańcuchem dostaw.

Komunikacja

Ilość zbieranych i przechowywanych w systemach zewnętrznych informacji koniecznych do planowania jest coraz większa. Wzajemne zależności między organizacjami także stają się coraz silniejsze. Konieczne jest więc zastosowanie zaawansowanej technologii komunikacyjnej, która pozwoli planować, optymalizować i realizować zadania w całym łańcuchu dostaw. ***Oracle Applications*** dostarcza szeroki zakres zaawansowanych narzędzi komunikacyjnych.

Oracle Applications pozwala na stosowanie transakcji EDI, które pozwalają z kolei na komunikację między partnerami handlowymi. Bramka EDI może zautomatyzować działania jak otrzymywanie przewidywań dotyczących zapotrzebowania od klientów, przesyłanie harmonogramów do dostawców czy otrzymywanie informacji o zaawansowanych działaniach transportowych.

Oracle Applications zawiera także zestaw produktów zaprojektowanych z myślą o instalacji systemu na platformie WWW. Moduły Web Customers, Web Suppliers i Web Employees zostały zaprojektowane jako łatwe do zainstalowania i wykorzystania aplikacje samoobsługowe. Z pomocą np. aplikacji ***Web Store*** i ***Web Configuration*** klienci mogą zamawiać samodzielnie dowolne produkty, nawet jeśli są to złożone konfiguracje zamówień. Wystarczy uruchomić standardową przeglądarkę internetową. Klienci mogą także mieć dostęp do własnych przewidywań na zapotrzebowanie, zmieniać je, uzyskiwać informacje o stanie zamówienia. Dostawcy mogą mieć dostęp do swoich harmonogramów. Wszystko jest należycie zabezpieczone i skonfigurowane tak, aby umożliwić jedynie autoryzowany dostęp.

Wszystkie aplikacje z serii ***Oracle Applications*** są dostępne w wersjach dla platformy WWW, zaprojektowane zgodnie z zasadami architektury ***Oracle Network Computing Architecture*** (NCA). Pozwala to nie tylko na tanie i szybkie instalacje wewnętrzne, ale także bezpieczne wejście z zewnątrz (przez linie telefoniczne) do dowolnych aplikacji, z pomocą standardowych przeglądarek internetowych. ***Oracle Applications*** wyposażono także w zestaw interfejsów API. Architektura API pozwala na integrację z systemami zewnętrznymi i pozwala na rozszerzanie systemu o nowe narzędzia i aplikacje. API pozwalają także firmie Oracle i jej klientom wdrażać najlepsze strategie architekturalne w swojej klasie.

Globalne deklaracje dostawy zamówień

Deklaracje dostawy zamówień są krytyczną częścią składową planowania i optymalizacji łańcucha dostaw. Popyt ze strony klienta musi zostać dynamicznie dopasowany do najbardziej efektywnego i zyskownego źródła dostaw już w momencie otrzymania zamówienia. ***Oracle Applications Supply Chain*** i opcja Supply To Promise pozwalają na realizację tej potrzeby. Możliwe jest określenie autoryzowanych źródeł dostawy dla danego klienta czy produktu lub ich kombinacji, a także standardowych metod dostawy i czasów tranzytowych. W momencie wprowadzenia zamówienia na ekranie prezentowana jest dostępność towaru z wszystkich autoryzowanych źródeł. Firma może zdeklarować się co do dostawy towaru w oparciu o dostępność materiałów czy mocy produkcyjnych.

Z pomocą ***Oracle RHYTHM*** możliwe jest utworzenie kontrolowanego ograniczeniami planu, który symultanicznie dopasuje popyt, materiały i moce produkcyjne i będzie ciągle dostosowywał plan do zmieniającej się dynamiki łańcucha dostaw. Ponieważ oprogramowanie bierze pod uwagę wszystkie elementy dynamiczne określające rzeczywisty całkowity czas dostawy, w rezultacie otrzymujemy dokładny, biorący pod uwagę wszystkie ograniczenia plan, tworzący podstawę deklaracji realistycznej daty dostawy.

Zaawansowane systemy wykonawcze

System planowania i optymalizacji łańcucha dostaw nie może być widziany jako prosty moduł lub mały, samodzielny system. Planowanie i optymalizacja łańcucha dostaw to połączenie wszystkich procesów przedsiębiorstwa oraz systemów koniecznych do planowania i realizacji planu w całym łańcuchu. Możliwość wykonania planu ogranicza jego wartość. Na jakość planu ma także wpływ czas cyklu wykonawczego. Wysoce niezawodne

systemy wykonawcze redukują ryzyko związane z planem. Szybkie systemy wykonawcze redukują horyzont czasowy, do którego dostosowuje się krytyczne decyzje, a więc redukują liczbę zmiennych do których trzeba dostosować plan. Szybkie wykonanie planu daje w rezultacie także szybko dostępne informacje zwrotne, co oznacza szybszą reakcję na wymagania i szybsze poprawki.

Oracle Applications to zaawansowany zestaw w pełni zintegrowanych aplikacji. Oracle dostarcza zaawansowane rozwiązania do planowania łańcucha dostaw, zaawansowane rozwiązania optymalizacyjne a także zaawansowane systemy wykonawcze. ***Oracle Applications*** zapewnia wsparcie dla pełnego zakresu produkcyjnych środowisk heterogenicznych od produkcji projektowej poprzez składanie na zamówienie, powtarzalną produkcję dyskretną aż po produkcję procesową. ***Oracle Applications*** także dostarcza jedynego, w pełni zintegrowanego systemu do obsługi otoczenia produkcji potokowej.

Oracle Applications to także wiele narzędzi do zaawansowanych systemów wykonawczych. Jako przykład, aplikacje realizacji zakupu generują w pełni zautomatyzowany, obsługujący przepływ zadań, pozbawiony dokumentów papierowych plan zakupu i opłat. *Oracle Supplier Scheduling* automatyzuje proces konwersji wewnętrznego planowania i informacji wykonawczych na harmonogramy transportowe dla dostawców i harmonogramy planistyczne. Te i inne zaawansowane narzędzia wykonawcze stanowią uzupełnienie i umacniają zaawansowane narzędzia do planowania i optymalizacji łańcucha dostaw w ***Oracle Applications***. Wszystkie razem, stanowią one najpełniejszy i najbardziej efektywny spośród dostępnych na rynku system obsługujący cały zakres zarządzania łańcuchem dostaw.

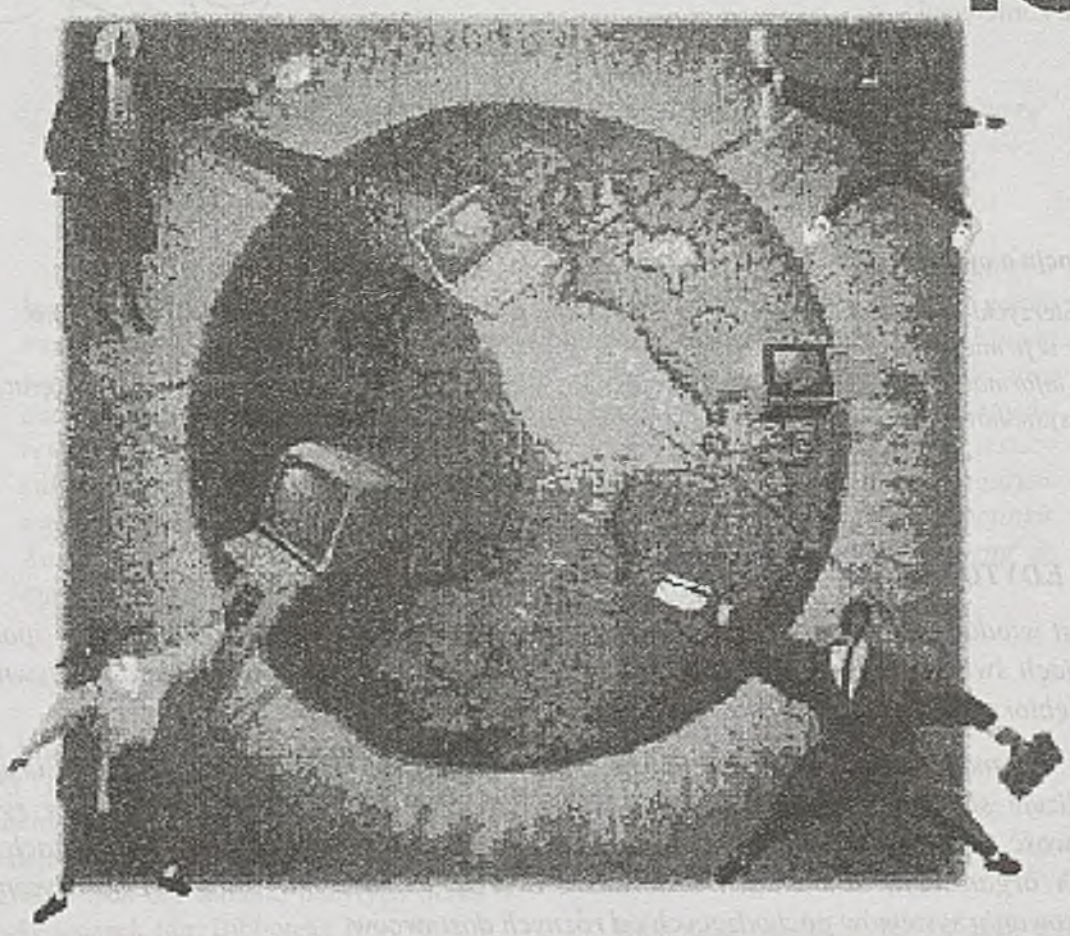
LITERATURA

1. Brandenburger, A. And Nalebuff, B. "Co-opetition" New York.: Doubleday, 1996
2. Goldratt, E. And Fox, B. The Race, North River Press: New York, 1986.
3. Hall, R. "Attaining Manufacturing Excellence" Homewood.: Dow Jones-Irwin, 1987.
4. Hammer, M. "Reengineering Work: Don't Automate, Obliterate." *Harvard Business Review*, May-June 1993, pp. 87-98.
5. Orlicky, J. "Material Requirements Planning; The New Way In Life in Production and Inventory Management" New York.: McGraw Hill, 1975.
6. Poirer, C. And Reiter, S. "Supply Chain Optimization" San Francisco.: Berret-Koehler, 1996.
7. Schonburger, R. "Japanese Manufacturing Techniques; Nine Hidden Lessons in Simplicity" New York.: Free Press, 1982.
8. Wight, O. "Production and Inventory Management in the Computer Age" Boston.: Cahners, 1974.

Stefan Sterzycki

ICL Poland Sp. z o.o.
ul. Leszno 21,
01-199 Warszawa
www.icl.pol.pl
Stefan Sterzycki – tel. (0-22) 833-31-06
stefan_sterzycki@x400.icl.co.uk
stefanst@polbox.com

Poinformowane społeczeństwo



ICL Poland Sp. z o.o.
Ul. Leszno 21, 01-199 Warszawa
Tel. (0-22) 631 05 66
632 22 95
fax. 632 09 79
www.icl.pol.pl

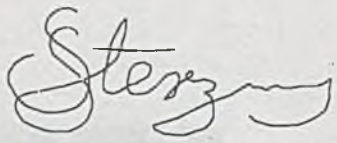
Obecne możliwości techniczne stwarzane przez informatykę uprawniają nas do dyskusji o rewolucyjnych zmianach jakie może nieść informatyka w codziennym życiu społeczności, także w życiu politycznym. Na pierwszy rzut oka wizja społeczności w której wszyscy mogą korzystać z dobrodziejstw dostępu do informacji i dzięki technologii informacyjnej realizować idee samorządności lokalnej (i nie tylko) wygląda na wizjonerską lub utopijną do czego zamierzam przekonać moich słuchaczy.

Ponieważ jednak informatyka nie rozwija się bez wstrząsów, cel jakim jest Dobrze Poinformowane Społeczeństwo może pozostać z boku. Moje wystąpienie będzie poświęcone omówieniu czynników sprzyjających i zagrożeń dla realizacji tego celu.

Materiały zostaną wręczone uczestnikom w czasie konferencji.

W przygotowaniu mojego wystąpienia opierałem się na szerokich badaniach ankietowych opracowanych na zlecenie ICL w sierpniu tego roku w Stanach Zjednoczonych Anglii, Francji, Szwecji i Niemczech oraz na efektach programów pilotażowych prowadzonych w Europie pod auspicjami Parlamentu Europejskiego, a także na dokumencie rządu brytyjskiego "Government green paper" oraz dyskusji prowadzonych na dorocznych konferencjach technicznych organizowanych przez ICL z udziałem głównie ekspertów niezależnych

Z poważaniem



Informacja o autorze

Stefan Sterzycki jest specjalistą od integracji systemów informatycznych w ICL Poland Sp. z o.o. Pracuje w firmie od 6 lat.
Lat 45; informatyk i ekonomista, specjalizujący się w systemach bankowości detalicznej i integracji dużych systemów informacyjnych.

NOTA EDYTORSKA

***ICL** jest wiodącą europejską firmą technologii informacyjnych. Posiada oddziały w ponad 70 krajach świata. Dostarcza zintegrowane systemy i usługi, które poprawiają efektywność przedsiębiorstw i organizacji.*

ICL Poland jest jednym z najbardziej dynamicznych oddziałów firmy w Europie. Specjalizuje się w integracji systemów informatycznych dla wybranych segmentów rynku (bankowość, przemysł, administracja publiczna, handel), kompleksowych rozwiązaniach dla wielkich organizacji, szkoleniach i doradztwie oraz świadczeniu usług specjalistycznych i serwisowaniu systemów pochodzących od różnych dostawców.

KONTAKT:

Stefan STERZYCKI ICL Poland SP. Z o.o. (22) 833-31-06
stefan_sterzycki@x400.icl.co.uk
stefanst@polbox.com

Anna Wyciechowska, Piotr Smorga

Max Elektronik S.A.
Ul. Mariacka 2
65-073 Zielona Góra
tel. (068) 327 00 32-33
Anna Wyciechowska - annaw@max.com.pl
Artur Smorga – arturs@max.com.pl

Aplikacje Max Elektronik w Bibliotece Śląskiej,

działające na platformie Sun Microsystems i PROGRESS Software

Funkcjonowanie tradycyjnej biblioteki opiera się na bazie katalogów kartkowych, w których ewidencjonowane są całe jej zbiory. Każdy czytelnik zanim weźmie do rąk upragnioną książkę rozpoczyna jej poszukiwanie wertując godzinami szafy z katalogami, w których umieszczane są karteczki z opisem bibliograficznym danej pozycji księgozbioru. Potem następuje żmudne odręczne wypisywanie stosów rewersów z sygnaturami. Pomyłka w wypełnieniu takiego zamówienia przedłuża jeszcze cały proces wypożyczania, gdyż po długim oczekiwaniu rewers powraca z magazynu z adnotacją magazyniera o błędnej sygnaturze. I wszystko zaczyna się od początku. Koszmar, nieprawdaż? Ale już wkrótce tak skomplikowana procedura dostępu do informacji i poszerzania wiedzy należeć będzie do przeszłości.

Od kilku lat szereg bibliotek w Polsce zmienia swój charakter, tworząc katalogi komputerowe. Polega to na założeniu, że zaprzestaje się wypełniania tradycyjnych kart bibliograficznych dla ostatnio nabytych przez bibliotekę książek (np. Biblioteka Narodowa). Niektóre placówki wdrażają nawet kompleksowe systemy wspomagające zarządzanie biblioteką, tzn. gromadzenie zbiorów, katalogowanie i ich udostępnianie. Wśród wszystkich tych jednostek zdecydowanie wyróżnia się Biblioteka Śląska w Katowicach. Przesądza o tym jej nowoczesność i stopień skomputeryzowania.

O bibliotece...

Biblioteka Śląska w Katowicach jest najstarszą biblioteką na terenie Górnego Śląska. Powołano ją do życia na przełomie lat 1922 i 1923 jako Bibliotekę Sejmu Śląskiego. W latach trzydziestych uzyskała samodzielność, stając się biblioteką naukową o charakterze publicznym. Taki status zachowała do dziś.

Biblioteka Śląska zawiera księgozbiór uniwersalny (ok. 1,3 mln woluminów, docelowo 2 mln) ze szczególnym uwzględnieniem wydawnictw śląskich i piśmiennictwa z zakresu nauk humanistycznych, ekonomicznych i społeczno-politycznych. Gromadzi i opracowuje całokształt krajowej produkcji wydawniczej w oparciu o tzw. egzemplarz obowiązkowy, piśmiennictwo obce i inne dokumenty służące rozwojowi badań naukowych oraz potrzebom kształcenia; udziela pomocy informacyjno-bibliograficznej instytucjom i poszczególnym osobom; inicjuje, prowadzi i publikuje prace naukowo-badawcze z zakresu nauki o książce i bibliotekoznawstwa, a zwłaszcza dotyczących Śląska; współpracuje z różnymi instytucjami w kraju i za granicą w celu wzajemnej wymiany materiałów bibliotecznych i doświadczeń. Udostępnia zbiory szerokiej publiczności, w szczególności pracownikom nauki i studentom.

Problemy z magazynowaniem książek

Biblioteka swoją dotychczasową siedzibę w Katowicach przy ul. Francuskiej otrzymała w ukończonym w 1934 roku budynku Domu Oświatowego, gmachu obliczonym na pomieszczenie najwyżej 100 tys. tomów, a nie ponad miliona zgromadzonych tam i w kilku innych miejscach. Wraz z rozrastaniem się księgozbioru biblioteki powstało pytanie, gdzie przechowywać coraz to nowe tomy, mając do dyspozycji jedynie magazyn o niewielkiej powierzchni. Dynamicznie wzbogacany księgozbiór spowodował, iż o konieczności budowy większego gmachu dyskutowano już w okresie międzywojennym.

Z biegiem lat sytuacja biblioteki stale się pogarszała. Na magazyny zbiorów zamieniono także sale, które wcześniej pełniły zupełnie inne funkcje np. czytelnie. Księgozbiór przechowywany był w różnych miejscach: w gmachu głównym, Oddziale Zbiorów Specjalnych w Katowicach oraz w Oddziale w Bytomiu i Cieszynie, co powodowało nawet kilkudniowe oczekiwanie na książki i kolejki w wypożyczalni.

Drastycznie pogarszały się warunki przechowywania księgozbioru. Wstrząsy, tąpnięcia (szkody górnicze), zalania, przeciążenia stropów ponad normy dopuszczalne sprawiły, iż bibliotece groziła całkowita zagłada.

W obliczu tej sytuacji środowisko humanistyczne Górnego Śląska postanowiło podjąć wyzwanie budowy nowej biblioteki. W 1991 r. zawiązano społeczny komitet, na czele którego stanął prof. dr hab. Jan Malicki, ówczesny dziekan Wydziału Polonistyki Uniwersytetu Śląskiego, a obecny dyrektor Biblioteki Śląskiej.

Pomysł wyjścia z sytuacji

Biblioteka Śląska w zakresie udostępniania zgromadzonych pozycji bibliograficznych miała dwa główne problemy, tj. brak swobodnego i szybkiego dostępu do swoich zbiorów i poważne trudności w zapewnieniu właściwej jakości obsługi czytelnika, co było spowodowane m.in. rozproszeniem księgozbioru. Głównym celem zamierzonej innowacji miało być przede wszystkim „udrożnienie" magazynów, tzn. możliwość szybkiego zlokalizowania żądanej pozycji księgozbioru i dostarczenia jej do określonego miejsca przy zminimalizowaniu czynności biurokratycznych związanych z rejestracją czytelnika i procesem wypożyczania.

I wtedy właśnie pojawiła się informacja o istnieniu zrobotyzowanej Biblioteki Publicznej w Bordeaux. Urządzenie zamontowane we Francji okazało się jednak bardzo kosztowne i nieefektywne - obsługiwało tylko niewielką część tamtejszych zbiorów. Robot, przesuwając się po szynach poprowadzonych między dwoma wysokimi regałami, umożliwiał dostęp do niewielkiej liczby książek, umieszczonych w pojemnikach ustawionych w jednym rzędzie. W efekcie robot okazał się mało wydajny i drogi w eksploatacji.

Spowodowało to, że zaczęto poszukiwać tańszego rozwiązania, polegającego na zastosowaniu zrobotyzowanego magazynu wysokiego składowania na wzór przechowywania części zamiennych m.in. w przemyśle motoryzacyjnym i AGD.

Nowy gmach

Aby ten kontrowersyjny pomysł wcielić w życie, niezbędnym okazało się wybudowa-

nie nowego budynku, który by pomieścił wszystkie zbiory biblioteki w jednym miejscu.

O nowym kształcie Biblioteki Śląskiej przesądził konkurs SARP rozstrzygnięty w czerwcu 1989 roku. Wybrano projekt zgłoszony przez Architektoniczne Atelier Realizacyjne ARAR: Juranda Jareckiego, projektanta prowadzącego i współautorów - Marka Gierlotkę i Stanisława Kwaśniewicza.

Nową bibliotekę zaprojektowano na planie kwadratu. Jej twórcy czerpali z tradycji modernizmu i symbolizmu lat 70. Projektanci uniknęli przytłoczenia skalą obiektu mimo ogromnej kubatury (81 tys. m. sześc.) i powierzchni użytkowej (17 tys. m. kw.). Bryła budynku jest zwarta i harmonijnie wtapia się w otoczenie. Budynek Biblioteki Śląskiej stoi w bardzo dogodnym miejscu, w centrum Katowic, obok osiedla Paderewskiego.

Nowy gmach posiada bogatą infrastrukturę. Dysponuje znakomicie wyposażoną audiowizualną salą audytoryjną (odczytowo-kinową) na 120 miejsc, z projekcją multimedialną sterowaną z tabletu oraz dwiema kabinami tłumaczeń. Jedenaście czytelni specjalistycznych zdolnych pomieścić naraz 220 czytelników oraz 52 km ciągów półek w magazynach zrobotyzowanych, zwartych oraz ogólnie dostępnych zapewni czytelnikom łatwy dostęp do informacji o różnym przeznaczeniu, nie tylko naukowym czy dydaktycznym.

Biblioteka dysponuje bogatą mediateką, najnowocześniejszym systemem konserwacji zbiorów, zintegrowanym systemem zarządzania i sterowania eksploatacją techniczną gmachu, ochroną przeciwpożarową (argon, azot), ochroną przeciwwłamaniową z kontrolą dostępu i instalacją telewizji użytkowej.

Samoobsługowy parking o 170 miejscach postojowych, krytych i otwartych dla samochodów osobowych, z dostępem dla niepełnosprawnych i bezpośrednim połączeniem z biblioteką oraz dobre zaplecze gastronomiczne z pewnością wpłynie na wysoki komfort korzystania z tego „przybytku wiedzy", jakim jest biblioteka.

Własna awaryjna kotłownia uniezależnia obiekt na wypadek braku dostaw ciepła z miejskiej sieci, a wysoka izolacyjność budynku wpływa na ekonomiczne ogrzewanie. Pełne zabezpieczenie konstrukcji przed destrukcją spowodowaną szkodami górniczymi występującymi w mieście osiągnięto stosując podział obiektu na 14 oddylatowanych segmentów. Istnieje także możliwość automatycznego nawilżania zieleni.

Unikalność rozwiązania

Zasadniczo biblioteki udostępniają swoje zbiory tylko za pośrednictwem czytelni lub wypożyczalni, pozwalając samodzielnie czytelnikom szukać książek wśród wolno stojących regałów. To metoda charakterystyczna dla bibliotek akademickich, wprowadzana w nowo budowanej Bibliotece Uniwersytetu Warszawskiego. System swobodnego dostępu do półek, uważany przez niektórych za najwygodniejszy, działa bez usterek tylko wtedy, gdy korzystający z biblioteki pozostawiają przeczytane książki na specjalnych stolikach, z których następnie zabierane są przez pracowników biblioteki i odkładane na właściwe miejsce. Jednak w praktyce wiele osób próbuje samodzielnie porządkować zbiory, wprowadzając w ten sposób spore zamieszanie na półkach. Codziennie, po zamknięciu biblioteki, trzeba ponownie porządkować zbiory, a do tego potrzebny jest liczny personel. Ponadto przy takim systemie wiele książek ginie, czasami bezpowrotnie. Dlatego też Biblioteka Śląska „postawiła" głównie na magazyn wysokiego składowania, do którego czytelnicy nie będą mieli bezpośredniego dostępu.

W nowo wybudowanym gmachu mieści się jedyny księgozbiór na świecie, którego spora część udostępniana jest za pośrednictwem zautomatyzowanego i skomputeryzowanego robota-podajnika. Plan ten zrealizowano umieszczając zbiory w jednym miejscu oraz przeprowadzając kompleksową informatyzację biblioteki.

Nowością jest magazyn wysokiego składowania zintegrowany z bazą biblioteczną PROLIB i system automatycznego transportu książki. To innowacja techniczna, unikalna nie tylko w skali naszego kraju, ale i Europy - podobne rozwiązanie zastosowano jedynie

w bibliotekach narodowych w Londynie i Paryżu.

Wprowadzenie tego kosztownego rozwiązania uzasadnione jest nie tylko zwiększeniem komfortu pracy czytelników, ale głównie wiąże się to z ogromnymi oszczędnościami dotyczącymi odsunięcia kapitałochłonnej rozbudowy magazynu o 20-30 lat. Inwestor wziął również pod uwagę bieżące oszczędności. Konstrukcja budynku wznoszonego w warunkach szkód górniczych, jak ma to miejsce w Katowicach, jest o 30% droższa niż w innych rejonach kraju. Lepiej więc zbudować jeden budynek z drogim wyposażeniem niż dwa z tanim.

Nowatorski system informatyczny

Magazyn wysokiego składowania znajduje się na ostatnim piętrze biblioteki (budynek ma 9 kondygnacji – dwie podziemne, parter i sześć pięter). Stąd jest bowiem najłatwiejszy dostęp do wszystkich czytelni i wypożyczalni, gdyż droga z góry na dół i odwrotnie jest najkrótszą z możliwych, jaką może przebyć zamówiona książka.

Umiejscowienie magazynu na samym dole gmachu bibliotecznego, pod ziemią, w warunkach szkód górniczych, wysokiego poziomu wód gruntowych i płytko poprowadzonej infrastruktury miejskiej, byłoby bardzo kosztowne i trudne z powodów konstrukcyjnych. Poza tym w przypadku pożaru, zbiory zgromadzone w części szczytowej biblioteki będą gaszone gazem obojętnym, który po stłumieniu ognia, jako cięższy od powietrza, znajdzie ujście na zewnątrz przez śluzy w podłodze magazynu.

W magazynie wysokiego składowania, obsługiwanym przez roboty, opartym o informatyczny system dostaw i ciągłą transmisję książki, bałagan będzie praktycznie niemożliwy. A zawiera on fragment zbiorów obejmujący najczęściej wypożyczane książki i docelowo ma pomieścić 600 tys. pozycji spośród ponad 1.300 tys. tomów.

W magazynie tym zainstalowano urządzenia austriackiej firmy TGW: robot–podajnik, przenośnik rolkowy i układ pojemników na książki. Wykorzystane w bibliotece austriackie rozwiązanie okazało się nieomal dwa razy bardziej wydajne niż jego francuski odpowiednik. Bowiem w systemie, ostatecznie zainstalowanym w Bibliotece Śląskiej, pojemniki z książkami magazynowane są aż w pięciu rzędach, na regałach o wysokości prawie dziewięciu metrów każdy. W każdym pojemniku natomiast znajduje się mniej więcej trzydzieści tomów, każdy w kopercie opatrzonej kodem kreskowym. Poznańska firma informatyczna Emax przygotowała tę instalację na potrzeby biblioteki, tworząc oryginalny system LIBROMAG. Jego twórcy przyznają jednak, że jeżeli nastąpiłby błąd w programie albo książka zostałaby włożona przez magazyniera do pojemnika o innym kodzie kreskowym niż wcześniej wprowadził do systemu, to dany tytuł mógłby się odnaleźć już tylko przez przypadek.

LIBROMAG jest w pełni zintegrowany poprzez oprogramowanie POMOST z kompleksowym systemem zarządzania biblioteką PROLIB. Te dwa ostatnie rozwiązania powstały w firmie Max Elektronik S.A. z Zielonej Góry. Bez PROLIB-a magazyn wysokiego składowania nie miałby sensu. Tylko biblioteka o całkowicie skomputeryzowanym katalogu może w pełni korzystać z tego typu magazynu.

PROLIB

Wdrażany w wielu bibliotekach uczelnianych, specjalistycznych, centralnych i publicznych Kompleksowy System Zarządzania Biblioteką PROLIB jest programem w całości napisanym przez polskich informatyków, przy ścisłej współpracy z bibliotekarzami. Kilka lat tej współpracy zaowocowało zintegrowanym pakietem pozwalającym na pełną automatyzację procesów bibliotecznych związanych z gromadzeniem, opracowaniem, wyszukiwaniem i udostępnianiem materiałów bibliotecznych, zapewniającym kontrolę czytelników oraz umożliwiającym komunikację (np. poprzez INTERNET) z zasobami innych bibliotek.

PROLIB jest profesjonalnym oprogramowaniem o wysokich walorach użytkowych, zdolnym przetwarzać duże ilości informacji, z odpowiednio zorganizowaną ochroną baz danych. Obsługa za pomocą klawiszy

funkcyjnych jest maksymalnie uproszczona. System ten zaprojektowano w oparciu o najnowsze w świecie narzędzia informatyczne (technologia CASE) i wykonano w języku czwartej generacji PROGRESS.

Sposób działania i cechy funkcjonalne systemu opierają się na cechach PROGRESS'a, a także na własnych cechach systemu PROLIB. Są to przede wszystkim: bezpieczeństwo, otwartość, przyjazność i rozwojowość.

System PROLIB zbudowany jest z modułów połączonych w bloki funkcjonalne, co umożliwia odpowiedni dobór funkcji oprogramowania dla potrzeb danej biblioteki bez konieczności zakupu pełnego oprogramowania. Poszczególne moduły można eksploatować niezależnie od siebie lub w ramach zintegrowanego systemu, a raz wprowadzona informacja jest wykorzystywana w całym systemie.

MODUŁY PODSTAWOWE:

- *Katalogowanie Wydawnictw Zwartych*
- *Katalogowanie Wydawnictw Ciągłych i Czasopism*
- *Wypożyczalnia*
- *OPAC*
- *Administrator*

MODUŁY DODATKOWE:

- *Gromadzenie Dokumentów*
- *Bibliografia*
- *Patenty*
- *Dokumenty Normalizacyjne*
- *Dokumenty Techniczno - Handlowe*
- *Import danych Przewodnika Bibliograficznego BN*
- *Obsługa bazy Słownika Języka Haseł Przedmiotowych BN w formacie USMARC*
- *Obsługa bazy BN „Czasopisma Nowe, Zawieszone, Zmieniające Tytuł"*
- *Centralny Katalog Czasopism Zagranicznych w polskich bibliotekach medycznych*
- *Tezaurus medyczny MeSH*
- *Import danych w formacie USMARC*

MODUŁY PROJEKTOWANE I ROZWIJANE:

- *Dokumenty audiowizualne*
- *Katalogowanie starodruków*
- *Katalogowanie druków muzycznych*
- *Obsługa bazy BN „POL-ARKA"*
- *Klient Z39.50*
- *Wypożyczalnia międzybiblioteczna*
- *Finanse i księgowość biblioteki*

Komputery w Bibliotece

System PROLIB w zakresie modułów podstawowych, został zakupiony przez Bibliotekę Śląską we wrześniu 1994 r. W 1996 roku firma MAX ELEKTRONIK S.A. została wybrana przez komisję przetargową do realizacji kompleksowej komputeryzacji w nowym gmachu Biblioteki Śląskiej. Początkowo, w starym budynku praca odbywała się w sieci UNIX na 16 terminalach, w grudniu 1996 r. MAX ELEKTRONIK rozszerzył sieć o kolejne 24 stanowiska, instalując także komputer SUN Sparc Server 1000E.

Przedsięwzięcie to obejmowało wykonanie sieci komputerowej, dostawę sprzętu oraz instalację, pełne wdrożenie i uruchomienie wraz ze stałą opieką serwisową następujących systemów:

- *Kompleksowy System Zarządzania Biblioteką PROLIB dla ok. 180 użytkowników.*
- *Oprogramowanie POMOST organizujące współpracę pomiędzy systemem PROLIB, a oprogramowaniem LIBROMAG (EMAX-Poznań), sterującym magazynem wysokiego składowania.*
- *Pakiet Microsoft Office dla ok. 150 użytkowników.*
- *Aplikacja obsługująca bazę danych Biblioteki za pomocą strony WWW w sieci Internet, stworzona w oprogramowaniu narzędziowym PROGRESS WebSpeed Transaction Server.*

- *Aplikacja - Informatorium Biblioteki Śląskiej, będąca multimedialnym wydawnictwem – przewodnikiem poświęconym historii, działalności i organizacji Biblioteki Śląskiej, którego zawartość w postaci tekstu i interaktywnych ilustracji, zdjęć, filmów i dźwięku, jest udostępniana przez INTERNET oraz lokalnie z komputerów PC i z kiosków informatycznych.*
- *System sieciowego udostępniania CD - ROM na serwerze NEXUS Tower.*
- *Moduły udostępniania baz danych Biblioteki Narodowej („Przewodnik Bibliograficzny", „POL-ARKA", „Słownik Języka Haseł Przedmiotowych BN", „Czasopisma Nowe, Zawieszone, Zmieniające Tytuł").*
- *Moduł opracowania starodruków.*

W lutym 1997 r. Biblioteka Śląska została włączona do sieci INTERNET, od tego też momentu dostępny jest w INTERNECIE katalog OPAC dla zbiorów biblioteki. W 1997 roku zakupiono także nowy moduł BIBLIOGRAFIA, który będzie wykorzystywany jako narzędzie do tworzenia „Bibliografii Śląskiej".

Sercem systemu informatycznego Biblioteki Śląskiej jest komputer SUN Ultra Enterprise 3000, mający za zadanie obsługę, bez spadku wydajności pracy, 300 aktywnych użytkowników, w tym - zdalnych, łączących się poprzez sieć INTERNET. Nad całością czuwa system operacyjny SOLARIS 2.6, a serwerem bazy danych jest PROGRESS Enterprise Server 7.x. Oprócz komputera głównego, jako serwer internetowy pracuje komputer SUN NETRA, który nie tylko zapewnia bibliotece komunikację ze światem zewnętrznym i dodatkowo umożliwia zaprezentowanie strony WWW BIBLIOTEKA ŚLĄSKA, ale także chroni bazę danych systemu PROLIB przed niepowołanym dostępem z zewnątrz (Firewall). Dodatkowym, rezerwowym komputerem jest SUN Sparc Server 1000E.

Większość z około 200 użytkowników pracuje w sieci na komputerach PC, wyposażonych w procesory PENTIUM, pod kontrolą systemu WINDOWS 95. Dopełnieniem wyposażenia jest blisko 80 różnego typu drukarek igłowych i laserowych.

Projekt sieci komputerowej obejmującej wszystkie piętra budynku, zakłada wykorzystanie elementów aktywnych firmy 3COM oraz okablowania strukturalnego mającego własności i spełniającego wymagania transmisyjne Kategorii 5 (zgodnie z normą EIA 568).

Kolejnym elementem projektu, zapewniającym jego niezawodność jest bezprzewodowy zasilacz awaryjny Silcon DATAPOWER DP 340C/10 o mocy 40 kVA. To trzyfazowe urządzenie zapewnia podtrzymanie napięcia przez 10 minut przy maksymalnym obciążeniu. Specjalne rozwiązanie konstrukcji baterii bezobsługowych i gazoszczelnych jest całkowicie bezpieczne dla otoczenia. Odpowiedni program pozwala na prawidłowe zakończenie pracy systemu w przypadku długotrwałej awarii napięcia.

Jak to działa ???

Do niedawna w Bibliotece Śląskiej w Katowicach czytelnik otrzymywał niektóre książki nawet dopiero po tygodniu. Z chwilą otwarcia nowego gmachu biblioteki 23 października br. i udostępnienia zbiorów pierwszym czytelnikom, większość pozycji księgozbioru trafi do rąk zamawiającego po około 5 minutach. To prawdziwy przewrót w dostępie do słowa pisanego.

Czytelnik chcąc wypożyczyć daną książkę przeprowadza kwerendę w katalogu OPAC na ekranie terminala umieszczonego w bibliotece, lub przeszukuje zbiory przez INTERNET. Jeśli jest zarejestrowanym użytkownikiem, może złożyć elektroniczny rewers. W przypadku, gdy książka znajduje się w magazynie wysokiego składowania, czytelnik, zamawiając książkę, sam uruchamia robota. Kiedy zamówienie na książkę dotrze przez sieć komputerową do jednego z pięciu podajników zwanych Mustangami, przesuwa się on po szynach do wskazanego obszaru magazynu i podnosi się na odpowiednią wysokość. Następnie wysuwa pięć połączonych ze sobą pojemników z książkami, odkłada na bok zbędne, zapamiętując przy tym ich tymczasowe położenie, a właściwe pudełko przenosi na rolkowy pas transmisyjny, którym dociera ono do stanowiska magazyniera. Osoba ta identyfikuje potrzebną książkę za pomocą elektronicznego czytnika kodów kreskowych,

po czym wyjmuje zamówioną pozycję z pojemnika i przekłada ją do innego, umieszczonego na szynach windy. Zamówione książki dostarczane są do czytelni poprzez system transportu wewnętrznego UNICAR, który wykorzystuje specjalne urządzenie telelift - specjalny wagonik, który poruszając się w pionie i poziomie po szynie błyskawicznie (3 m/s) dociera do odpowiedniej czytelni lub wypożyczalni. Taka organizacja pracy biblioteki zmienia diametralnie szybkość korzystania z wiedzy. Jeżeli dystans, jaki ma do przebycia zamówiona pozycja wyznaczony jest najbardziej odległymi od siebie punktami biblioteki i magazynu, to cały ten proces trwa 7 minut. W większości przypadków jednak książka trafi do rąk czytelnika w ciągu 5 minut. Centralny komputer optymalizuje bowiem pracę systemu na podstawie prowadzonej przez siebie statystyki. Wskazuje najintensywniej eksploatowane pojemniki, które trafiają do pierwszego, najłatwiej dostępnego szeregu w regale.

Jeżeli zaś książka jest przechowywana w magazynach zwartych przesuwnych (kilka kondygnacji z przesuwnymi regałami kompaktowymi znajduje się w dolnej części głównego trzonu budowli), to na tę pozycję trzeba będzie czekać być może nieco dłużej, a zależy to od czasu, w jakim magazynier dotrze do regału, zidentyfikuje książkę i wróci z nią na miejsce, gdzie na przesyłkę będzie już czekał odpowiednio zaprogramowany telelift.

Problemy przy wdrażaniu

Głównym problemem informatyzacji Biblioteki Śląskiej jest jak najszybsze wypełnienie bazy danych, tak aby w momencie przeniesienia księgozbioru do nowego obiektu, przynajmniej ta jego część, która zostanie umieszczona w zautomatyzowanym magazynie, była opracowana w systemie komputerowym. Poważnym problemem są tutaj braki kadrowe. Biblioteka dysponuje zbyt małą ilością pracowników, którzy mogą wprowadzać dane, a ponadto są oni obciążeni bieżącymi obowiązkami oraz pracami związanymi z przeprowadzką. A przecież szybkie wprowadzenie do bazy danych o księgozbiorze jest strategicznym warunkiem powodzenia całego przedsięwzięcia.

Wobec powyższego, dużym ułatwieniem przy tworzeniu bazy danych opisów bibliograficznych jest opcja systemu pozwalająca na import danych w formacie MARC/BN z Przewodnika Bibliograficznego BN. Dokonano także konwersji Słownika Języka Haseł Przedmiotowych BN, zakupionego przez bibliotekę w postaci dyskietek.

Oczekiwane korzyści z innowacji

Zwrócenie kosztów inwestycji informatycznych w przypadku instytucji biblioteki dokonuje się w sposób pośredni i długofalowy. Trudne jest także zbadanie i wyliczenie, w jaki sposób i w jakiej mierze uzyskanie szybkiej i aktualnej informacji o księgozbiorze, miejscu przechowywania danego dokumentu, możliwości jego wypożyczenia oraz o jego zawartości rzeczowej przekłada się na wymierne korzyści finansowe. W przypadku biblioteki jako organizacji niedochodowej należałoby raczej zwrócić uwagę na korzyści płynące z szybkości uzyskania informacji, co zazwyczaj dopiero w dalszym etapie zapewnia dochody finansowe.

Oczekiwanym rezultatem inwestycji będzie na pewno ułatwienie i przyspieszenie środowisku naukowemu regionu dostępu do własnych zbiorów oraz do zbiorów i informacji będących w posiadaniu innych bibliotek. Przeprowadzona innowacja zapewni usprawnienie obsługi czytelnika. Wprowadzenie skomputeryzowanych katalogów uniezależni czytelnika od bibliotekarza. Będzie on mógł sam zaplanować kolejność czytania książek i moment, w którym po nie sięgnie.

Przeprowadzone wdrożenie przyniesie także poprawienie komfortu i efektywności pracy bibliotekarza, który - skracając czas, jaki musiał przeznaczyć na wszystkie formalności związane z obsługą, ewidencją i statystykami – będzie mógł poświęcić więcej uwagi m.in. poprawie jakości obsługi czytelnika. Bibliotekarz przestanie być urzędnikiem, a stanie się przewodnikiem, opiekunem i doradcą. Zdaniem prof. dr hab. Jana Malickiego, dyrektora Biblioteki Śląskiej, bibliotekarze wykorzystując ten zintegrowany system, będą natychmiast wiedzieć, jakie są

nowe nabytki biblioteki i będą mogli od razu z nich skorzystać. Administracja zaś zyska pełną kontrolę nad zakupami od chwili zamówienia książki czy czasopisma aż do zapłacenia faktury. Zmieni to radykalnie model zarządzania biblioteką, która upodobni się do przedsiębiorstwa, wykorzystującego systemy klasy ERP.

To tyle założeń i rozważań na temat przyszłego funkcjonowania Biblioteki Śląskiej. A jak w praktyce wywiąże się ona z pokładanych w niej nadziei, okaże się już jesienią, kiedy to czytelnicy co dzień będą sprawdzać jej użyteczność i postępowość.

Korzyści z informatyzacji Biblioteki Śląskiej

1. Gromadzenie, wyszukiwanie i porządkowanie informacji w ramach elektronicznego katalogu (dostępnego także poprzez interfejs WWW) umożliwiającego generowanie zestawień bibliograficznych według różnorodnych kryteriów.
2. Unowocześnienie sposobu magazynowania książek poprzez zbudowanie skomputeryzowanego magazynu wysokiego składowania, zintegrowanego z elektroniczną bazą wydawnictw PROLIB, co wraz z zastosowaniem teleliftu (sterowanego elektronicznie systemu transportu książki) umożliwia automatyzację zamówień magazynowych i automatyczne dostarczenie książki do czytelnika.
3. Skrócenie czasu dostępu do zbiorów (do 5-7 min.) i usprawnienie obiegu książki na drodze magazyn-wypożyczalnia, magazyn-czytelnia, czytelnia-czytelnia.
4. Przyspieszenie przebiegu procesów bibliotecznych począwszy od wpływu książki do biblioteki, poprzez jej opracowanie, aż do udostępnienia czytelnikowi.
5. Elektroniczne księgowanie zbiorów biblioteki w momencie opracowywania wydawnictw (sumowanie wartości nabytków), zakończone wydrukiem ksiąg inwentarzowych bez potrzeby „ręcznego" ich wypisywania.
6. Zabezpieczenie materialne zbiorów biblioteki poprzez elektroniczny monitoring (fizyczne zabezpieczenie książek - paski magnetyczne, wydzielenie i kontrola stref czytelników i personelu)
7. Uporządkowanie procedury zakupów nowych książek (unikanie powtarzających się zakupów) dzięki modułowi PROLIBA „Gromadzenie".
8. Wydruk katalogowych kart bibliotecznych w systemie PROLIB. Pozwala to na wykorzystanie czasu pracowników biblioteki do innych zadań.
9. Umożliwienie importu do bazy danych - opisów bibliograficznych z baz danych zapisanych w różnych formatach i sporządzonych w innych bibliotekach. Proces aktualizacji i uzupełniania danych katalogu dokonuje się zatem elektronicznie, bez potrzeby ręcznego ich wpisywania.
10. Szybszy obieg informacji i usprawnienie wewnętrznej komunikacji pomiędzy działami i pracownikami biblioteki poprzez lokalną sieć komputerową.
11. Szerokie wykorzystanie poczty elektronicznej - duża część korespondencji bibliotecznej dotyczącej np. ofert wymiany książek z innymi bibliotekami, odpowiedzi na kwerendy biblioteczne itp.
12. Wprowadzenie systemu zamówień i rezerwacji książek Biblioteki Śląskiej, umożliwiającego lokalną i zdalną (przez INTERNET) realizację tych procesów.
13. Dostęp czytelników do INTERNETU - pozwala na korzystanie z pełnej informacji z różnych źródeł - zdalny dostęp do katalogów innych bibliotek.
14. Realizacja serwisu informacyjnego o bibliotece, tzw. Informatorium, tj. multimedialnego serwisu WWW, orientującego czytelnika w planie przestrzennym Biblioteki Śląskiej, prezentującego jej historię i bieżącą działalność oraz pracowników – Informatorium dostępne on-line, lokalnie w kioskach informatycznych oraz na płytach CD.

Jacek Pacholczyk

Novell Polska
Sienna 64, 00-825 Warszawa
Tel: +48 22 6203979
faks: +48 22 6203103
Jacek Pacholczyk, Area Director for Central Eastern Europe, Novell

Wpływ technologii informatycznych na funkcjonowanie jednostek i organizacji w świecie u progu XXI wieku"

George Orwell w swojej powieści „Rok 1984" opisał świat, w którym panuje jeden system: Wielki Brat nie śpi i stale obserwuje wszystkich obywateli, a policja może czytać w ich myślach — zamknięty świat konformizmu — wszechświat zamkniętych standardów...

Taki świat kojarzy się ze światem, w którym wszyscy korzystają z jednego rozwiązania. W swojej książce „Nowy wspaniały świat" Aldous Huxley przewidział pojawienie się zjawisk i elektronicznych gadżetów, których istnienie nie dziwi dziś nikogo. Niestety świat przedstawiony przez autora jest miejscem sterylnym, w którym nie ma miejsca na indywidualność.

Jednakże pracując w firmie Novell – teraz, w 1998 roku – możemy z zadowoleniem stwierdzić, że na szczęście obydwaj ci wielcy pisarze się mylili. Weszliśmy w wiek Internetu, wiek otwartych standardów, wiek, w którym osoby prywatne lub niewielkie firmy mogą opracowywać nowe technologie i w ciągu zaledwie kilku dni tworzyć przyszłość.

Novell, wraz z jego prezesem Erikiem Schmidtem, odgrywa w tym świecie zasadniczą rolę, **dostarczając rozwiązania oparte na otwartych standardach i przyczyniając się do realizowania idei prowadzenia interesów za pośrednictwem Internetu.**

Cieszymy się, że możemy dzisiaj zaprezentować Państwu rozwiązania, które ułatwią Wam poznawanie **Nowego Wspaniałego Świata Otwartych Standardów** i – dzięki wykorzystaniu możliwości Internetu – umożliwią utworzenie oszczędniejszego i bardziej konkurencyjnego środowiska służącego do prowadzenia interesów.

W dzisiejszym nieprzewidywalnym świecie, system NetWare 5 – stanowiący podstawę dla rozwiązań Novella na miarę XXI wieku – pośredniczy w prowadzonych za jego pomocą interesach. Korzysta z niezawodnej i sprawdzonej infrastruktury. Dzięki rozwiązaniom Novella usprawniającym działanie przedsiębiorstw — takim jak intranety, ekstranety, produkty służące do prowadzenia biznesu elektronicznego, hurtownie danych, rozwiązania ułatwiające współpracę i umożliwiające pracę zdalną — można spać spokojnie, wiedząc, że te najwyższej jakości produkty sprawiają, iż używająca ich firma bije na głowę swą konkurencję.

Technologia ta ma rewolucyjny wpływ na firmy. Przeniknęła do wszystkich gałęzi przemysłu i przyspieszyła tempo zachodzących w nich zmian. Można zaryzykować stwierdzenie, że większość z tych przedsiębiorstw nie mogłaby funkcjonować bez infrastruktury, którą zapewnia im ta technologia. Czy

w obliczu kolejnego stulecia Państwa firmy są przygotowane do pełnego wykorzystania tych możliwości? A — co ważniejsze — czy Państwo są do tego przygotowani?

Kilka wieków temu uważano, że Słońce, gwiazdy i planety są od siebie niezależne, a Ziemia jest płaska. Poglądy te można porównać z tradycyjnymi opiniami o sieciach komputerowych, w myśl których Internet nie jest w żadnym punkcie połączony z siecią lokalną przedsiębiorstwa.

Dziś zobaczą Państwo, w jaki sposób Novell zmienia ten pogląd.

Państwa cały sieciowy wszechświat...

.... rozpoczyna się tutaj. I teraz !!!

Oferujemy gotowe produkty i rozwiązania już dziś — nie składamy fałszywych obietnic. Novell stanowi pomost między panującymi w przeszłości poglądami na temat konstrukcji wszechświata sieci, a dzisiejszymi przekonaniami — zajmuje pozycję w centrum sieciowego kosmosu.

Dzięki Novellowi, poznawaniu nowego świata nie towarzyszą żadne niebezpieczeństwa. Niezależnie od tego, czy jest to nadchodzący rok 2000, wydajność przedsiębiorstwa czy potrzebne firmie rozwiązania służące do prowadzenia interesów za pośrednictwem Internetu, Novell oferuje bezpieczną podstawę do prowadzenia biznesu elektronicznego i stałego rozwoju przedsiębiorstwa. Warto zauważyć, że ten nowy świat jest po prostu uaktualnionym obrazem starego świata.

Wraz z nadejściem każdej technologii nowej generacji wydaje się, że możliwości przedsiębiorstw znacznie wzrosną. W miarę ewoluowania stosowanych rozwiązań — od systemów opartych na komputerach mainframe przez systemy klient-serwer i rozwiązania sieciowe — technologia przyspieszała tempo zmian zachodzących w firmach, dając im nowe możliwości. W chwili powstania Internetu narodziły się całkowicie nowe sposoby komunikowania się i prowadzenia interesów.

Korzystanie z nowych metod prowadzenia biznesu elektronicznego nie oznacza porzucenia istniejącej bezpiecznej infrastruktury ani opracowanych procedur, lecz rozbudowanie ich i szersze wykorzystanie ich możliwości. Podstawowe cechy stosowanych do tej pory sieci oraz mechanizmów ułatwiających kontrolę i zapewniających spójność można z łatwością przenieść do rozwiązań obecnych w nowym świecie. Od dzisiaj, od tej chwili mogą Państwo uważać Internet za otwarty i bezpieczny dla Państwa przedsiębiorstw.

Wiek Internetu oznacza dla Państwa i dla Waszych firm nowe i zadziwiające możliwości.

Analitycy IDC oceniają, że wartość transakcji przeprowadzonych za pośrednictwem Internetu w 1997 roku wyniosła około 12 miliardów dolarów. Przewiduje też, że kwota ta wzrośnie do 400 miliardów dolarów w 2002 roku.

W swojej prognozie firma Forrester zgadza się z tymi oszacowaniami twierdząc, że do 2002 roku wartość transakcji z udziałem indywidualnych klientów wzrośnie z 4 do 26 miliardów dolarów, a wartość transakcji przeprowadzanych bezpośrednio między przedsiębiorstwami wzrośnie z 16 miliardów dolarów do 268 miliardów.

Liczba połączeń internetowych rośnie. Co miesiąc w Internecie przesyła się tysiące terabajtów danych, a liczba ta podwaja się co 90 dni

Przedsiębiorstwa instalują intranety. Liczba instalowanych intranetów gwałtownie wzrosła w 1997 roku. W ciągu następnych 5 lat 60% wszystkich firm korzystających z intranetów zainstaluje ekstranet

Jak jednak to osiągnąć??? W jaki sposób włączyć się do gry??? W jaki sposób wykorzystać możliwości, które zapewnia Nowy Wspaniały Świat, i zwiększyć obroty i zyski swojej firmy???

Trzeba podjąć działania, aby nie pozostać w tyle. Należy spojrzeć na świat z otwartym umysłem i w całkowicie nowy sposób. Pomyślcie Państwo o tym.

Gdzie znajduje się centrum Internetu? Do tego pytania należałoby jeszcze dodać „intranetu i ekstranetu". Pytanie to wydaje się być łatwe, lecz nie ma na nie łatwej

odpowiedzi. Większość osób, którym je zadano, wyglądało na zaskoczonych. Niektórzy odpowiadali, że chodzi o firmy Yahoo, Netscape lub Amazon. Inni, że to Cisco lub inni producenci infrastruktury sieciowej stanowią centrum Internetu.

W rzeczywistości zawsze podczas korzystania z dowolnej przeglądarki internetowej — niezależnie od miejsca, z którego się łączymy — nasuwa się inne pytanie: nie „gdzie leży środek", lecz KTO stanowi centrum tej sieci? Dla nas odpowiedź jest łatwa — to Państwo stanowią centrum Sieci! Wszystko zależy od Państwa — intranet Państwa firmy, używany tam ekstranet i praktycznie wszystkie mechanizmy dostępne w Internecie. Siła tych rozwiązań zależy od poszczególnych użytkowników. Proszę sobie wyobrazić Internet bez indywidualnych użytkowników korzystających z niego!

Spróbujmy rozwinąć tę linię rozumowania. Co widzą Państwo podczas logowania się do sieci, przed przyłączeniem się do niej? To ekran logowania systemu NetWare, nieprawdaż??? Jeśli indywidualni użytkownicy stanowią centrum Internetu, a w ich biurach zainstalowano system NetWare Novella, to jakie wobec tego miejsce zajmuje firma Novell, jej system NetWare, ponad 80 milionów użytkowników jej produktów i 4 miliony instalacji sieciowych? Właśnie centralne miejsce w Internecie!!! Państwo i sieci, które stosujecie, znajdują się w centrum wszechświata sieci komputerowych. Niezależnie od tego, czy jest to sieć lokalna, ekstranet czy intranet, Państwo znajdują się w centrum.

Firma Novell i jej system NetWare oferują możliwość bezpiecznego prowadzenia interesów z klientami i dostawcami zarówno wewnątrz firmy, jak i na zewnątrz — jednokierunkowo lub w obie strony.

Wprowadzając na rynek system NetWare 5 ponownie rewolucjonizujemy sposób myślenia, gdyż udowadniamy, że system ten stanowi centrum wszechświata sieci komputerowych.

Do tej pory przedsiębiorstwa traktowały zainstalowane sieci lokalne, Internet, intranety i ekstranety jak oddzielne elementy. Sieć jest jednak trwałym fundamentem, na którym są oparte wszystkie inne elementy. **Sieć to Novell! A teraz jest to system NetWare 5!**

NetWare 5, najnowsza wersja wiodącego na świecie sieciowego systemu operacyjnego, korzysta wyłącznie z protokołów IP, udostępnia otwarte standardy, oferuje niezależność od stosowanej platformy i stanowi bezpieczną podstawę dla tworzenia systemów internetowych. Jest to pierwszy produkt wyposażony w tak wydajną implementację języka JAVA na serwerze — standardowego narzędzia umożliwiającego prowadzenie interesów za pośrednictwem Internetu. Ułatwiamy programistom tworzenie aplikacji, które się staną niezawodną częścią rozwijających się systemów do prowadzenia interesów. **Udostępniamy serwer aplikacji opracowany dla nowego świata!**

W ciągu ostatnich 3–4 lat, wraz z rosnącą popularnością Internetu, wszyscy uświadomili sobie ogromne znaczenie tej sieci dla przedsiębiorstw. W tym samym czasie sieci komputerowe stały się podstawowym elementem systemów ułatwiających prowadzenie interesów. Obecnie należy zmodyfikować definicję sieci tak, aby obejmowała ona także Internet. **Na mapie rozwiązań sieciowych naniesiono poprawki. Od tej pory w jej centrum leży system NetWare 5.**

Prowadząc interesy w wieku informacji odkrywamy jednak pewien paradoks. Niedawno opublikowane w prasie raporty opisują słabe punkty systemu Windows NT, których istnienie narażało serwery WWW na ataki ze strony hakerów. Spowodowało to wzrost obaw o to, jak intensywnie można rozwijać biznes elektroniczny. Przedsiębiorstwa informują, że ponad połowa ataków na ich własność jest dokonywana za pośrednictwem Internetu. W raporcie opublikowanym niedawno przez firmę IDC oceniono, że tylko 16% wszystkich europejskich firm używa stron WWW do sprzedawania swoich produktów. Zarządy wielu firm powoli uświadamiają sobie fakt, że informacje, na przykład dane zgromadzone przez przedsiębiorstwo, stanowią obecnie najcenniejsze zasoby firmy.

Aby w pełni wykorzystać te zasoby, należy je udostępnić pracownikom w całej firmie oraz

partnerom na zewnątrz firmy. Są nimi dostawcy, klienci, współpracownicy i firmy świadczące usługi. Prowadzi to do paradoksu — jak udostępnić najcenniejsze zasoby firmy, zabezpieczając równocześnie ich spójność i chroniąc je przed dostępem osób niepowołanych.

Ścisłe zintegrowanie Internetu z istniejącymi systemami służącymi do prowadzenia interesów jest tylko częścią możliwości systemu NetWare 5. Równie ważne są oferowane przez ten produkt mechanizmy przydatne w przedsiębiorstwie. Oferuje on bezpieczeństwo, łatwość zarządzania i wysoką wydajność. Cechy te sprawiają, że system ten stanowi podstawę dla rozwiązań Novella na miarę XXI wieku i spełnia wszystkie potrzeby najbardziej wymagających przedsiębiorstw.

Opracowując system NetWare 4, Novell stworzył najbezpieczniejszą sieć dostępną na rynku. Jest to jedyny produkt na świecie opatrzony – wydawanym przez rząd Stanów Zjednoczonych – certyfikatem C2 Red Book, oznaczającym najwyższy poziom bezpieczeństwa sieci komputerowej. Mechanizmy te zostały rozbudowane przez NDS i NetWare 5. Usługi katalogowe zapewniają łatwość zarządzania wszystkimi elementami sieci, kontrolowania ich i uzyskiwania do nich dostępu z jednego centralnego punktu. Dzięki nim administratorzy mogą łatwo i sprawnie konfigurować prawa dostępu i nadzorować ich przestrzeganie. Oznacza to, że tylko autoryzowani użytkownicy — zarówno pracownicy firmy, jak i osoby z zewnątrz — mogą korzystać z dostępnych w sieci informacji i zasobów.

Novell jest pionierem w opracowywaniu bezpiecznych rozwiązań opartych na standardach i służących do prowadzenia biznesu elektronicznego. Najnowsze technologie są wyposażone w możliwość używania wyłącznie protokołu IP (*Internet Protocol*), korzystają z mechanizmu SSL (*Secure Socket Layer*), protokołu LDAP (*Lightweight Directory Access Protocol*) oraz obsługują standard bezpieczeństwa PKI (*Public Key Infrastructure*).

Standard PKI będzie stanowił podstawę dla prowadzenia transakcji, ponieważ oferuje on możliwość korzystania z unikatowych podpisów cyfrowych przeznaczonych dla osób i transakcji. Zaimplementowanie tego standardu na serwerze wraz z usługami NDS i systemem NetWare 5 dowodzi, że po raz kolejny Novell ustanawia obowiązujące standardy. Technologia ta jest niezbędna do bezpiecznego prowadzenia biznesu elektronicznego.

Wprowadzone przez firmę Novell oprogramowanie BorderManager — przeznaczone do zarządzania połączeniem między różnymi sieciami — podnosi poziom bezpieczeństwa, ponieważ umożliwia zarządzanie prawami dostępu indywidualnych użytkowników. Ponadto oprogramowanie to sprawdza informacje docierające z zewnątrz za pomocą zapór ogniowych poziomu I, II i III, zapewnia bezpieczeństwo dzięki korzystaniu z wirtualnych sieci prywatnych, a także niezrównaną szybkość dostępu do sieci WWW, uzyskaną dzięki zastosowaniu technologii proxy-cache. Jest to już taki poziom bezpieczeństwa, który umożliwia prowadzenie biznesu elektronicznego.

Mechanizmy bezpieczeństwa są podstawą wszystkich elementów wchodzących w skład rozwiązań Novella na miarę XXI wieku. Jest to pierwszy kompletny pakiet produktów sieciowych, dzięki któremu realne staje się prowadzenie biznesu elektronicznego.

Klienci mają teraz do dyspozycji inteligentną infrastrukturę internetową, która zintegruje wszystkie te usługi i umożliwi im zebranie mechanizmów zarządzania siecią w jednym centralnym punkcie, który będzie odgrywał rolę bramki łączącej sieć lokalną ze światem zewnętrznym. **Sercem tego rozwiązania jest system NetWare 5.**

Zanim przedstawię te rozwiązania, powinniśmy porozmawiać o kosztach ich instalowania w przedsiębiorstwach. Firma IDC podaje, że obecnie ponad 76% kosztów implementowania technologii informatycznych, czyli całkowitych kosztów obsługi sieci, jest przeznaczanych na zarządzanie siecią, komputerami biurowymi, aplikacjami i użytkownikami.

Całkowite koszty obsługi sieci to jednak tylko jeden z kilku aspektów. O sieci należy

myśleć jak o rozwiązaniu umożliwiającym wykonanie znacznie większej liczby zadań, a jednocześnie pozwalającym na zmniejszenie koniecznych nakładów. Nie można również zapomnieć o korzyściach wynikających z wyższej wydajności pracy osiągniętej dzięki sieciom, o wartości, jaką to rozwiązanie stanowi dla firmy, a także — co najważniejsze — o tym, że sieć ułatwia lub wręcz umożliwia odniesienie sukcesu. Na przykład, jak duże są straty ponoszone w Państwa firmach z powodu przestojów w działaniu sieci komputerowej??? Jeśli pracują Państwo w biurze maklerskim lub banku, to każda godzina wyłączenia sieci może kosztować miliony dolarów!!!

Idea Total Value Networking oznacza wdrażanie rozwiązań sieciowych, które podnoszą wartość firmy przez wpływanie na wzrost wydajności pracy, efektywności, konkurencyjności, a równocześnie pozwalają na obniżenie całkowitych kosztów obsługi sieci, co w rezultacie umożliwia rozwinięcie działalności i zwiększenie zysków.

Total Value Networking oznacza także traktowanie zasobów technologii informatycznych jako źródła wielu możliwości, a nie jako przyczyny ponoszenia dodatkowych wydatków.

Zasoby informatyczne podnoszą wartość firmy, gdyż umożliwiają wzrost wydajności pracy oraz konkurencyjności, co pozwala na obniżenie kosztów i zwiększenie zysków. Ułatwiają one prowadzenie handlu za pośrednictwem sieci, analizowanie i gromadzenie danych, współpracę i pracę zdalną, zarządzanie sieciami i komputerami biurowymi, a także — dzięki rozwiązaniom zwiększającym dostępność sieci — pozwalają na skrócenie czasu, w którym sieć nie działa.

Novell zajmuje wyjątkową pozycję na rynku, ponieważ jest jedynym dostawcą rozwiązań sieciowych, opartych na sprawdzonym i niezawodnym katalogu, które są dostępne już dziś. Dzięki prawie 40 milionom użytkowników, usługi NDS stały się standardem obowiązującym na rynku.

Zestaw rozwiązań Novella na miarę XXI wieku w pełni korzysta z tej unikatowej architektury katalogu opracowanej przez naszą firmę. BorderManager i serwery firmy Netscape na platformę NetWare umożliwiają przedsiębiorstwom szybkie tworzenie katalogów swoich produktów i udostępnianie ich w sieci. Pozwalają na budowanie środowiska służącego do przetwarzania poufnych danych finansowych. To prawdziwe przykłady rozwiązań służących do prowadzenia biznesu elektronicznego. System NetWare 5 stanowi mocny fundament dla rozwiązań służących do tworzenia i obsługi hurtowni danych. Obecnie jest dostarczany z najnowszą bazą danych Oracle8. Umożliwia bezpieczne przechowywanie informacji i udostępnianie ich autoryzowanym użytkownikom w celu przeprowadzenia analizy. Środowisko NetWare tradycyjnie jest najszybszą platformą dla aplikacji baz danych Oracle, teraz rozbudowano je o wirtualną maszynę Javy wprowadzoną w systemie NetWare 5, spełniającą funkcje serwera aplikacji napisanych w języku JAVA. Dzięki temu NetWare jest niewiarygodnie silną platformą umożliwiającą korzystanie z rozwiązań służących zarówno do prowadzenia biznesu elektronicznego, jak i do obsługiwania baz danych.

Współpraca oznacza możliwość zbudowania infrastruktury, za pomocą której pracownicy mogą się komunikować ze sobą i współużytkować informacje niezależnie od miejsca, w którym pracują. Dzięki mechanizmowi umożliwiającemu pracę zdalną mogą oni pracować w domu, hotelu, czy w biurze klienta. Wygląd biura przyszłości nie będzie przypominał tego, co widzimy dzisiaj. Praca za pośrednictwem sieci telefonicznej staje się jedną z możliwości oferowanych przez środowiska pracy przyszłości. Jest to możliwe dzięki systemowi GroupWise.

Novell prowadzi prace nad stworzeniem całkowicie bezawaryjnego systemu oferując środowisko sieciowe o dużej odporności na błędy. Dzięki serwerowi StandbyServer i klastrom Novell High Availability Server for NetWare użytkownicy mogą być pewni, że podstawowe systemy będą dla nich zawsze dostępne. System ManageWise pozwala na zautomatyzowanie wielu czynności związanych z zarządzaniem siecią, dzięki czemu specjaliści mogą się skoncentrować na ważniejszych zadaniach. Firma IDC twierdzi, że administratorzy, którzy korzystają

z systemu ManageWise, mogą obsługiwać o 25% więcej komputerów biurowych i o 33% więcej serwerów. Dzięki niemu można było także obniżyć o 50% koszty związane z dotarciem do odległych działów firmy.

Komputer dostosowany do indywidualnych potrzeb, który potrafi automatycznie rozwiązywać pojawiające się problemy, sprawia, że klienci są bardziej zadowoleni z jakości obsługi. Możliwość szybkiego rozprowadzania oprogramowania pozwoli Państwu na zaoszczędzenie mnóstwa czasu i pieniędzy. Aby spełnić takie wymagania, Novell wprowadził na rynek oprogramowanie Z.E.N.works, które ułatwia zarządzanie komputerami biurowymi. Nazwa tego produktu oznacza pracę w sieci bez żadnego wysiłku ze strony użytkowników. Mają oni bardzo łatwy dostęp do potrzebnych im informacji i zasobów wtedy, gdy tego potrzebują. Oprogramowanie to zapewnia pracownikom niewiarygodną swobodę i elastyczność, pozwalając administratorowi na efektywne zarządzanie siecią z centralnego punktu.

Uruchomiony niedawno przez Novella program PartnerNet łączy sprzedawców produktów, firmy tworzące kanały dystrybucyjne, konsultantów, producentów sprzętu, sprzedawców oprogramowania, dostawców usług, autoryzowane ośrodki szkoleniowe, firmy oferujące pomoc techniczną i inne. Od wysoko wykwalifikowanych ekspertów zajmujących się wszystkimi rozwiązaniami opracowanymi przez Novella, klienci mogą oczekiwać usług najwyższej jakości. Za pośrednictwem Internetu jest dostępny ClubInternet, który przez całą dobę umożliwia skontaktowanie się ze specjalistami Novella. W naszym programie DeveloperNet bierze udział ponad 25 tysięcy programistów z całego świata. Produkty, w tym wiele nowych aplikacji napisanych w języku Java, opracowane przez setki firm otrzymały od nas certyfikaty zgodności, umożliwiające ich używanie pod kontrolą systemu NetWare 5. Lista wszystkich tych przedsiębiorstw znajduje na stronie WWW DeveloperNet. Novell oferuje także usługi doradcze z zakresu metod migrowania z systemu NetWare, zarządzania komputerami biurowymi i innych. Opracowaliśmy również programy szkoleniowe poświęcone wszystkim produktom i usługom Novella, z których można skorzystać w jednym z 1000 ośrodków w Europie.

Przedstawiliśmy stary świat i podpowiedzieliśmy, w jaki sposób można sobie zapewnić wstęp do nowego świata i skorzystać z nowych możliwości! Ci z Państwa, którzy usiłowali prowadzić interesy za pośrednictwem Internetu, z pewnością natknęli się na wiele przeszkód. Bariery te zostały już usunięte. Nikt nie powinien zadawać pytania: Czy należy korzystać z technologii opracowanych przez Novella... Właściwe pytanie brzmi: **Kiedy?**

Czy mogą sobie Państwo pozwolić na to, aby nie skorzystać z tych możliwości? Tak jak wtedy, gdy Mikołaj Kopernik odkrył, że Słońce jest centrum naszego układu planetarnego...

... Tak my mamy nadzieję, że odkryli dziś Państwo, iż sieć Novella stanowi środek wszechświata sieci komputerowych. Jest to także centrum prowadzonych przez Państwa interesów!!! Wraz z rozwiązaniami Novella na miarę XXI wieku oferujemy Państwu bezpieczną i sprawdzoną platformę umożliwiającą prowadzenie interesów za pośrednictwem Internetu i osiąganie zysków. Już dziś! Chcielibyśmy prowadzić z Państwem długotrwałą i owocną współpracę, która powiedzie nas w rok 2000 i dalej!

Cała Państwa sieć i wszechświat Waszych interesów ROZPOCZYNA SIĘ TUTAJ I TERAZ!

Andrzej Maciej Wierzba

Epoka postindustrialna a informatyka

Przedsiębiorstwo socjalistyczno – industrialne

Polska lat socjalizmu była państwem z ideami epoki industrialnej. Celem nie było ani dobro jednostki, ani jej prawa, ale produkcja. Jako, że rynek nie narzucał jakościowych miar towarów, skupiono się na cechach ilościowych. Szczęście ludzi przeliczane na miliony ton wydobytego węgla czy wyprodukowanego cementu. Ludzie mieli być jak najbardziej zunifikowani do pracy w typowym biurze. Pięcioletnie plany produkcji z nakazowym przydziałem surowców, dewiz i rozdzielnikiem wyprodukowanych towarów tworzyły tzn. gospodarkę planową. I chyba właśnie to oddanie epoki socjalizmu ideałom epoki przemysłowej stało się przyczyną jego końca. Zmieniły się technologie, zmieniły się potrzeby ludzkie i sposoby zarządzania. Po to, aby istnieć dalej należało coś zmienić. I tym czymś był socjalizm.

Prowadzone w ostatnich 9 latach zmiany koncentrują się na wymianie ustawodawstwa i przywracaniu praw własności. To są niezbędne procesy, aby od scentralizowanego molocha przejść do gospodarki wolnorynkowej. Przykład wielu skomercjalizowanych przedsiębiorstw pokazuje, że nie wystarczy dodać na końcu S.A., aby z nieruchawego państwowego molocha stać się zwinnym drapieżnym tygrysem. Od praw własności ważniejszym staje się sposób zarządzania przedsiębiorstwa. W epoce industrialnej wzorcowym typem zarządzania był i jest nadal system hierarchiczny.

Prezes Zarządu rozdziela zadania wykonywane przez członków zarządu. Członkowie zarządu mają swoich dyrektorów, którzy stoją w hierarchii nad kierownikami. Kierownicy zarządzają majstrami, którzy nadzorują robotników. Odpowiedzi na pytanie, dlaczego ten system zarządzania istnieje nadal jest wiele. Po pierwsze to przyzwyczajenie. Dlaczego coś zmieniać i podejmować ryzyko skoro można wziąć bezpiecznie pewną pensję za robienie tego co zawsze. Koszty reorganizacji są zwykle duże i mogą się wiązać ze zwolnieniem kolegów. Nie wiadomo, jak zostanie to odebrane przez rynek, gdy na wizytówce zamiast -Dyrektor działu wdrożeń pojawi się Koordynator ds. wdrożeń. Po drugie skoro nasi klienci pracują w firmach zhierarchizowanych, to pewnie będą oczekiwali od nas takich samych struktur - trzech wicedyrektorów, asystentek, sekretarek i działu windykacji. Takich obiektywnych powodów jest wystarczająco dużo aby zachować status quo. Wewnątrz dużych organizacji ta nieefektywność jest mało widoczna. Skoro wszyscy wypełniają firmowe druki, piszą dobre plany i raporty zgodne z przyjętymi procedurami to są na tyle zajęci, że wyglądają na bardzo efektywnie pracujących.

Epoka postindustrialna

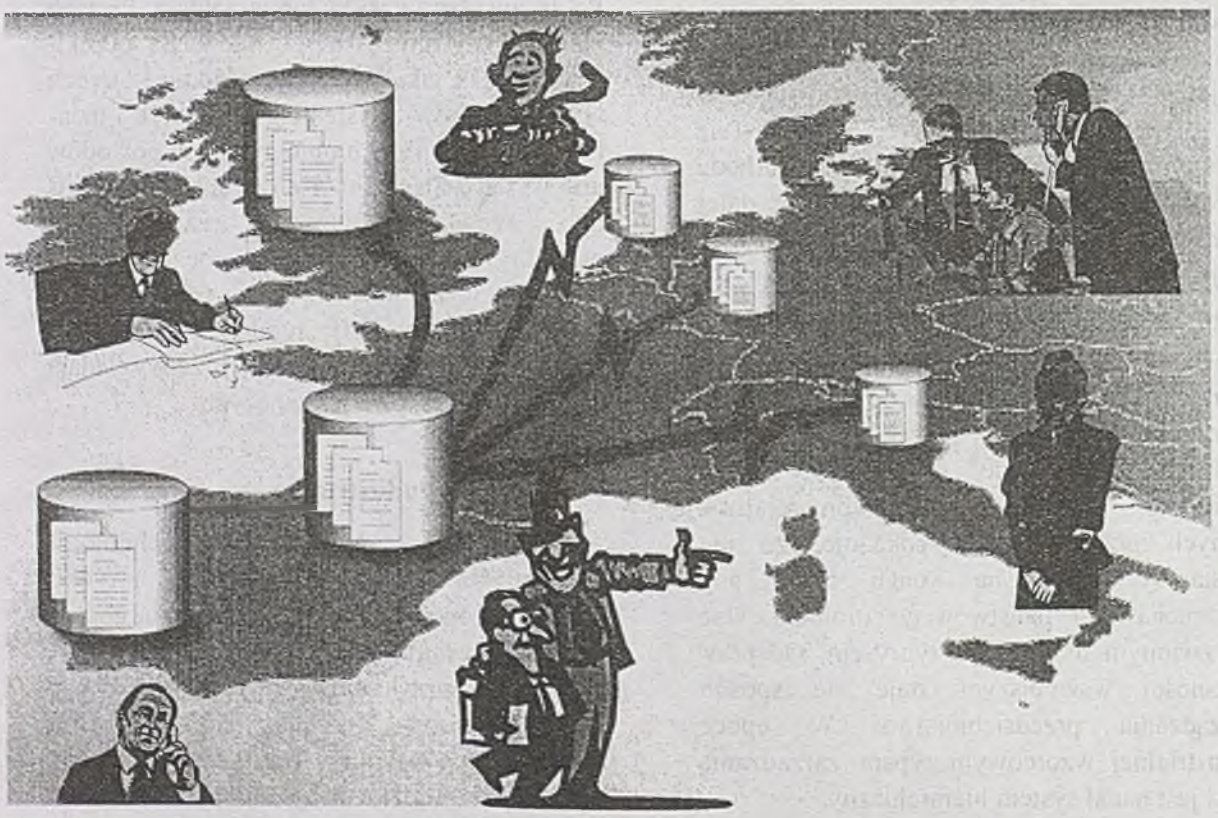
Pojawienie się nowych technologii i zapotrzebowanie ze strony rynku spowodowały rozwój nowych dziedzin gospodarki. Biotechnologie, telefonia komórkowa, internet są przykładami cudów rynkowych naszych czasów. Styl działania tych firm opiera się na szybkim i skutecznym ataku. Tradycyjny model zarządzania jest dla nich zupełnie niewydolny. O ile firma industrialna przypomina wojsko podążające na spotkanie wroga równymi szeregami w uprasowanych

mundurach, o tyle nowoczesna firma wygląda jak siły szybkiego reagowania - znienacka ląduje na terytorium wroga, szybko się przegrupowuje i tak samo szybko zajmuje teren swojej konkurencji. Aby większa organizacja mogła działać skutecznie tymi metodami zmianie muszą ulec podstawowe zasady gry. Najważniejszym celem stają się rozwój i przetrwanie firmy oparte na:

- dbałości o potrzeby i wymagania klientów
- stworzeniu pracownikom dobrych motywujących warunków pracy
- systemie efektywnie sterującym prowadzonymi działaniami.

Wygodnie jest przedstawić model nowoczesnej firmy w postaci trzech współśrodkowych pierścieni. Kolo utworzone przez te pierścienie jest figurą, która przy danej powierzchni posiada największy styk z otoczeniem - z klientami i partnerami, z którymi firma będzie współpracować. Zewnętrzny pierścień skupia najważniejszych pracowników firmy - tych, którzy mają najbliższy kontakt z klientami i z wyrobami produkowanymi dla klientów. Od tych pracowników zależy w największym stopniu jakość produktów oraz dbałość o potrzeby i wymagania klientów. Do zarządzania realizacją poszczególnych funkcji odpowiadających w tradycyjnej firmie działom – handlu, marketingu, dostaw powoływani są koordynatorzy. Mogą też być oni wyznaczani do realizacji długoterminowych zadań – np. do realizacji serialu telewizyjnego czy opracowania nowego modelu samochodu.

Na drugi pierścień składają się służby pomocnicze zajmujące się zaopatrzeniem, utrzymaniem biura, finansami, obsługą informatyczną. W centrum koła mieści się zarząd firmy analizujący sytuację wewnętrzną i zewnętrzną oraz skutecznie planujący strategię działań bieżących i rozwoju. Mniejsza liczba poziomów sprzyja dobrej komunikacji i zwiększeniu odpowiedzialności.



Informatyka a nowoczesna firma

Aby móc wprowadzić informatykę należy najpierw zmienić sposób zarządzania. Aby móc zmienić sposób zarządzania trzeba posługiwać się nowoczesną informatyką. Z tej niby-sprzeczności wynika, że nie da się szybko zreformować zarządzania w firmie i wprowadzić do niej nowoczesnej informatyki. Jest to długotrwały proces zmian organizacyjnych wspieranych przez właściwą informatykę. Firmy epoki przemysłowej różnią się od firm postindustrialnych i tak samo różnią się odpowiednie systemy informatyczne. Tak jak firma epoki przemysłowej różni się od firmy postindustrialnej tak samo różnią się ich systemy informatyczne. Ten pierwszy typ firmy posiada określoną strukturę, zakresy obowiązków i procedury. Zadaniem informatyków będzie więc zakodowanie tych wszystkich elementów w opracowywanym oprogramowaniu. Dzięki nim powstaje system najczęściej oparty o centralną bazę danych pozwalający na wypełnianie rutynowych dokumentów i otrzymywanie typowych raportów. W jego ramach obowiązkiem pracowników niższych poziomów jest wpisywanie danych, zadaniem pracowników wyższych poziomów jest czytanie raportów i wydawanie poleceń podwładnym.

System informatyczny dla firmy postindustrialnej musi uwzględniać jej specyfikę:

- firma wykonuje wiele różnych i często zmieniających się usług i produktów
- aby zapanować nad swoimi pracownikami musi posiadać dobry program finansowo-księgowy i spójny system controllingu
- codzienne decyzje podejmowane są przez wszystkich pracowników
- pracownicy często są poza firmą, ale muszą korzystać z jej zasobów informacji
- firma ma wiele oddziałów
- dla sukcesu niezbędne jest ciągłe pozyskiwanie i utrzymywanie klientów.

O ile w firmie industrialnej pracownik głównie wprowadza informacje, to w firmie postindustrialnej głównie korzysta z informacji. System informatyczny takiej wzorcowej firmy jest najczęściej rozproszony. Osobista baza danych zawiera informacje potrzebne do pracy np. agenta ubezpieczeniowego odwiedzającego z notebookiem swoich klientów. Dane w niej umieszczone pozwalają na sporządzenie umowy ubezpieczeniowej zgodnie z obowiązującymi aktualnie przepisami i stawkami. Korzystając z poczty elektronicznej albo sieci w biurze dane o sporządzonych polisach są replikowane do bazy w oddziale przedsiębiorstwa. Na notebooki przenoszone są dane o telefonicznych zgłoszeniach klientów dotyczących tego agenta ubezpieczeniowego i zmiany w wewnętrznych unormowaniach. Informacje niezbędne dla centrali są replikowane on-line do centralnej bazy przedsiębiorstwa. Korzystając z dostępnej technologii taki system może być wykonany tak samo łatwo, jak system scentralizowany. Kolejnym elementem systemu firmy postindustrialnej jest intranetowa baza wiedzy o firmie. Wytyczne, zarządzenia, okólniki, ogłoszenia, wzory dokumentów są cały czas dostępne dla pracowników w sieci komputerowej oddziału i poprzez replikację także na prywatnych notebookach. Pracownicy firmy postindustrialnej pracujący w wielu miejscach chętnie korzystają z poczty elektronicznej i danych dostępnych za pośrednictwem sieci internetowej. Ich przenośne komputery pozwalają na drukowanie potrzebnych dokumentów i bezpośrednie wysyłanie faksów.

Aby przedsiębiorstwa mogły przetrwać na globalnym, konkurencyjnym rynku muszą skutecznie wdrażać nowoczesne metody zarządzania wspomagane informatyką. Ta informatyka musi być traktowana jako nowe metody pracy, a nie jako nowo zakupione programy. Ten sposób pozwoli na ewolucję informatyki i stylu zarządzania w kierunku efektywnego, nowoczesnego przedsiębiorstwa.

Autor jest Prezesem Zarządu firmy Sysoft reprezentującej na polskim rynku Sybase Inc.

Jarosław Deminet

Czy istnieje życie poza komputerami?

Komputery stały się nieodłącznym elementem wszystkich dziedzin naszego życia i zaczynają stopniowo wypierać wiele innych urządzeń. Zniknęły maszyny do składu drukarskiego, zastąpione przez systemy typu desktop publishing, korzystające ze standardowych komputerów i drukarek. Nie projektuje się już precyzyjnych mechanicznych zegarków — zastępują je układy elektroniczne za kilka dolarów sztuka. Szlachetne wieczne pióra z irydowymi i platynowymi stalówkami, którymi wielcy pisarze pisali swe książki, ustępują przed programami edycyjnymi. Graficy wolą korzystać z myszy i odpowiedniego programu, niż z piórka i węgla. Ostatnio nie buduje się już nawet tuneli aerodynamicznych do badania własności projektowanych samochodów i samolotów — zamiast nich korzysta się z symulacji komputerowej.

Czy to wszystko oznacza, że postęp techniczny będzie się coraz bardziej koncentrował w branży komputerowej? Że postęp ten będzie się sprowadzał do szybszego procesora, większej pamięci i lepszej drukarki? Że za kilka lat każde urządzenie techniczne będzie się składało z komputera z dołączonymi urządzeniami peryferyjnymi? Że praca inżyniera będzie miała sens tylko wówczas, gdy będzie projektował lepszy procesor i szybszą drukarkę?

SPIS TREŚCI

PREZENTACJE INSPIRE Str.

1. **Tomasz Byzia** (InfoViDE) *„Narzędzia zapewnienia i kontroli jakości oprogramowania"* 1
2. **Prof. Piotr Dembiński** *„Metody formalne w projektowaniu oprogramowania: mity i rzeczywistość"* 5
3. **Dr Janusz Zalewski** *„Systemy czasu rzeczywistego - projektowanie i aspekty praktyczne"* 17
4. **Grzegorz Kazimierski** (UMBRELLA PROJECT) *„Wdrażanie systemów jakości zgodnych z ISO 9000"* 29
5. **Prof. Henryk Krawczyk** *„Modele i ocena jakości systemów informatycznych"* 31
6. **Adam Czerniński** *„Zarządzanie przez jakość. Rola testowania w zarządzaniu budową systemu w cyklu Architected RAD"* 47
7. **Borys Stokalski** (InfoViDE) *„Zasady audytu przedsięwzięć informatycznych"* 51

WYKŁADY MONOGRAFICZNE

1. **Prof. Jan Zabrodzki** *„Grafika komputerowa"* 65
2. **Bogusław Jackowski** (BOP) *„Znaki na niebie, ziemi i gdzie indziej"* 67
3. **Dr Tomasz Traczyk** *„Bazy danych, narzędzia dostępu do danych i hurtownie danych"* 75

PREZENTACJE

1. **Grzegorz Janczewski** (XEROX) *„Nowa strategia XEROX - technologia cyfrowa od biurka do masowej produkcji dokumentów"* 87
2. **Jakub Połeć** (MICROSOFT) *„Windows NT 4.0. Terminal Serwer Edition"*. 89
3. **Wojciech Sosnowski** (LOTUS) *„Zarządzanie wiedzą przy użyciu narzędzi pracy grupowej"* 101
4. **Paweł Adamek, Mirosław Tarczoń** (Océ) *„Kompleksowe rozwiązania w kopiowaniu i drukowaniu"* 103
5. **Pierre Sabloniere** (IBM) *„Systemy ERP na platformie IBM S/390."* *„IBM S/390 – Konsolidacja serwerów"* 107
6. **Andrzej Wyrzykowski** (MHM COMPUTER) *„Enterprise Storage - gwarancja bezpieczeństwa danych"* 111
7. **Dariusz Duchowski, Krzysztof Zabielski** (EMAX) *„Najnowsze rozwiązania w dziedzinie przetwarzania dokumentów"* 113
8. **Dr Robert Rylko (Computer Associates)** *„Zarządzanie stacjami roboczymi w ramach zintegrowanego systemu zarządzania środowiskiem IT"* 115
9. **Dr Ryszard Krawczyński** (ORACLE) *„Oracle Applications - Planowanie łańcucha dostaw"* .. 119
10. **Stefan Sterzycki** (ICL) *„Poinformowane społeczeństwo"* 129
11. **Anna Wyciechowska, Piotr Smorga** (Max Elektronik) *„Aplikacje Max Elektronik w Bibliotece Śląskiej, działające na platformie Sun Microsystems i PROGRESS Software"* 131
12. **Jacek Pacholczyk** (NOVELL) *„Wpływ technologii informatycznych na funkcjonowanie jednostek i organizacji w świecie u progu nowego tysiąclecia"* 139

DYSKUSJE PANELOWE

1. **Andrzej Maciej Wierzba** *„Epoka postindustrialna a informatyka"* 145
1. **Jarosław Deminet** *„Czy istnieje życie poza komputerami ?"* 149

„Problemy jakości i niezawodności środków informatyki"

pod redakcją Piotra Fuglewicza

Wydawca:

Polskie Towarzystwo Informatyczne
Oddział Górnośląski
40-012 Katowice
ul. Św. Jana 10

Druk i oprawa:

Zakłady Mechaniczne „BUMAR-ŁABĘDY" S.A.
Drukarnia Informatyki
44-109 Gliwice 9
ul. Mechaników 9

Przygotowanie materiałów do druku:

Mirosława Minich

LISTA SPONSORÓW
CZTERNASTYCH JESIENNYCH SPOTKAŃ PTI
Mrągowo, 16-20 listopada 1998r.

COMPAREX Systemy Informatyczne Sp. z o.o. 01-059 Warszawa, ul. Okopowa 47

Computer Associates Sp. z o.o. 00-697 Warszawa Centrum LIM. Al. Jerozolimskie 65/79

EMAX Sp. z o.o. 61-894 Poznań, ul. Niezłomnych 1c

Hewlett Packard Polska Sp. z o.o. 02-222 Warszawa, Al. Jerozolimskie 181

IBM Polska Sp. z o.o. 02-134 Warszawa, ul. 1 Sierpnia 8

ICL Poland Sp. z o.o. 01-199 Warszawa, ul. Leszno 21

InfoViDE Sp. z o.o. 02-119 Warszawa, ul. Pruszkowska 17

Lotus Development Polska 02-134 Warszawa, ul. 1 Sierpnia 8

Max Elektronik S.A. 65-073 Zielona Góra, ul. Mariacka 2

MHM Computer Polska Sp. z o.o. 00-895 Warszawa, ul. Biała 4

Microsoft Sp. z o.o. 00-844 Warszawa, ul. Grzybowska 80/82

NOVELL Polska Sp. z o.o. 00-825 Warszawa, ul. Sienna 64

„Océ-Poland Ltd." Sp. z o.o. 02-232 Warszawa, ul. Łopuszańska 53

ORACLE Polska Sp. z o.o. 00-833 Warszawa, Sienna Center, ul. Sienna 75

United Nations Development Programme UMBRELLA PROJECT 00-608 Warszawa
Al. Niepodległości 186

XEROX Polska Sp. z o.o. 00-762 Warszawa, ul. Belwederska 20/22

CZTERNASTE JESIENNE SPOTKANIA PTI MRĄGOWO '98 – 16-20 LISTOPADA 1998

PROBLEMY JAKOŚCI I NIEZAWODNOŚCI ŚRODKÓW INFORMATYKI

DATA	PREZENTACJE PRZEDPOŁUDNIOWE			WYKŁADY	PREZENTACJE POPOŁUDNIOWE		DYSKUSJE PANELOWE
	9.00-9.45	9:45-10:30	11:00-11:45	12:00-13:30	15:15-16:00	16:15-17:00	17:30-19:00
PN 16.XI	Tomasz Byzia (INFOVIDE) *Narzędzia zapewnienia i kontroli jakości oprogramowania* INSPIRE	Tomasz Grabowski (IBM) *Niezawodne systemy przetwarzania w dużych systemach baz danych na przykładzie DB2*	Grzegorz Janczewski (XEROX) *Nowa strategia XEROX – technologia cyfrowa od biurka do masowej produkcji dokumentów*	Prof. Piotr Dembiński *Metody formalne w projektowaniu oprogramowania: mity i rzeczywistość* INSPIRE	Jakub Połeć (Microsoft) *Windows NT 4.0 Terminal Server Edition*	Wojciech Sosnowski (Lotus) *Zarządzanie wiedzą przy użyciu narzędzi pracy grupowej*	Dr Janusz Zalewski *Systemy czasu rzeczywistego – projektowanie i aspekty praktyczne* INSPIRE
WT 17.XI	Grzegorz Kazimierski (Umbrella Project) *Wdrażanie systemów jakości zgodnych z ISO 9000* INSPIRE	Paweł Adamek, Mirosław Tarczoń (OCE) *1. Kopiowanie i drukowanie zdecentralizowane oraz zastosowanie dokumentów kolorowych we współczesnym biurze.* *2. Druk wysokowydajny w przedsiębiorstwie – tendencje rozwojowe.*		Prof. Henryk Krawczyk *Modele i ocena jakości systemów informatycznych* INSPIRE	Pierre Sabloniere (IBM) *1. Systemy ERP na platformie IBM S/390* *2. IBM S/390 – Konsolidacja Serwerów*		*Zarządzanie przez jakość* Moderator: Adam Czerniński INSPIRE
ŚR 18.XI	Andrzej Wyrzykowski (MHM Computer) *Enterprise Storage – gwarancja bezpieczeństwa danych*	Dariusz Duchowski, Krzysztof Zabielski (EMAX) *Najnowsze rozwiązania w dziedzinie przetwarzania dokumentów*	Dr Robert Ryłko (Computer Associates) *Zarządzanie stacjami roboczymi w ramach zintegrowanego systemu zarządzania środowiskiem IT*	Borys Stokalski *Zasady audytu przedsięwzięć informatycznych* INSPIRE	Dr inż. Ryszard Krawczyński (ORACLE) *Oracle Applications – Planowanie łańcucha dostaw*		*Epoka postindustrialna a informatyka* Moderator: Maciej A. Wierzba
CZW 19.XI	Stefan Sterzycki (ICL) *Poinformowane społeczeństwo*	Waldemar Celej (Hewlett Packard) *Rozwiązania wysokiej dostępności*	Artur Smorga (MaxElektronik SA) *Aplikacje MaxElektronik w Bibliotece Śląskiej, działające na platformie SUN Microsystems i PROGRESS Software*	Prof. Jan Zabrodzki *Grafika komputerowa i wirtualna rzeczywistość – stan dzisiejszy*	Jacek Pacholczyk (Novell) *Wpływ technologii informatycznych na funkcjonowanie jednostek i organizacji w świecie u progu nowego tysiąclecia*		*Czy jest życie poza komputerami?* Moderator: Jarosław Deminet
PT 20.XI		Bogusław Jackowski *Tutorial: Znaki na niebie, zieni i gdzie indziej*		Dr Tomasz Traczyk *Bazy danych, narzędzia dostępu do danych i hurtownie danych*	W ŚRODĘ O 20:30 KONCERT KUBY SIENKIEWICZA		

PTI *Zarząd Główny*
ul. Żelazna 87
00-879 Warszawa
tel./fax (0-48-22) 652-32-59
pti_zg@orgmasz.waw.pl

Oddział Górnośląski
ul. Św. Jana 10
40-012 Katowice
tel./fax (0-48-32) 253-96-78
ptiog@netpol.pl

CZŁONKOWIE WSPIERAJĄCY POLSKIE TOWARZYSTWO INFORMATYCZNE

IBM

IBM Polska Sp. z o.o.
02-134 Warszawa, Wiśniowy Business Park, ul. 1 Sierpnia 8

océ

„Océ-Poland Ltd." Sp. z o.o.
02-232 Warszawa, ul. Łopuszańska 53

oraz

Centralny Ośrodek Informatyki Górnictwa S.A., 40-065 Katowice, ul Mikołowska 100

Fabryka Wtryskarek „PONAR-ŻYWIEC" S.A., 34-300 Żywiec, ul;. Stolarska 21

ICL Poland Sp. z o.o., 01-199 Warszawa ul. Leszno 21

InfoViDE Sp. z o.o., **02-119 Warszawa, ul. Pruszkowska 17**

Ster-Projekt Sp. z o.o., 02-652 Warszawa, ul. Magazynowa 1

Zakłady Przemysłu Barwników „BORUTA" S.A., 95-100 Zgierz, ul. Struga 30

International Business Systems, **51-124 Wrocław, ul. H Kamieńskiego 57**