

*Robert Ambroziak\**

## KONSTRUKCJA APLIKACJI WSPOMAGAJĄCEJ ZARZĄDZANIE

*W artykule zawarto krótki opis metodyki zarządzania klasy MRP/ERP, oraz przedstawiono przykładowe zależności technologiczne związane z konstrukcją systemu wspomagającego zarządzanie przedsiębiorstwem przemysłowym.*

*In this paper the MRP/ERP class management is presented. The example of structure and technical application of MRP/ERP system is also presented.*

### **Wprowadzenie**

W dzisiejszych bardzo konkurencyjnych czasach, wszelkie dobrze przemyślane działania w sferze zarządzania informacją, sprawna jej selekcja, oraz zapewnienie taniego do niej dostępu wpływają na rozwój firmy (przedsiębiorstwa). Żyjemy obecnie w wieku „informacji”, która daje nam przewagę, a jej brak może być powodem naszego opóźnienia, co w konsekwencji może prowadzić do klęski finansowej. Dzisiejszy „biznes” jest bardzo dynamiczny, nie pozwala nam stać w miejscu, zmusza nas abyśmy wyprzedzali nasze obecne działania. Od wielu lat duże korporacje inwestują w komputerowe systemy zarządzania, które integrują całą infrastrukturę informatyczną korporacji w jeden centralny system zarządzania. Często inwestycje te sięgają lub nawet przewyższają kwoty kilkudziesięciu milionów dolarów, zatem stają się nieosiągalną barierą konkurencyjności dla średnich lub małych przedsiębiorstw. Większość średnich i małych firm wykorzystuje pewne systemy wspomaganie zarządzania, lecz dotyczy to przede wszystkim programów księgowych, często nie udostępniających danych w sieci komputerowej, co uniemożliwia efektywne wykorzystanie tych danych. Pozostałe funkcje - rozliczanie kosztów, kadry i płace, środki trwałe, sprzedaż, gospodarka materiałowa, planowanie i zarządzanie produkcją, zarządzanie jakością, remonty i utrzymanie ruchu przedsiębiorstwa, zarządzanie przedsięwzię-

---

\* Katedra Informatyki Stosowanej, Politechnika Łódzka

ciami i planowanie inwestycji - stosowane są już o wiele rzadziej. Należy pamiętać, że im większa organizacja, tym trudniej kierownikom nad nią zapanaować, gdyż bardzo szybko rośnie stopień komplikacji działania, tym wyraźniejsza zachodzi potrzeba komputerowego wsparcia. Rozpatrując oddzielnie logistykę, marketing i finanse, popadając w nadmierną specjalizację, traci się dystans niezbędny do zarządzania. Kiedy czas i elastyczność decydują o powodzeniu, lub klęsce przedsięwzięć, zintegrowane systemy wspomagania zarządzania są narzędziem wprost wymarzonem. Przy rosnącej skali działania w pewnym momencie po prostu pojawia się konieczność wprowadzenia komputerowych systemów zarządzania, które zwiększają zdolność szybkiej adaptacji do potrzeb klientów i rynków zbytu, co w dzisiejszych czasach jest decydującym czynnikiem konkurencyjności. Z często przeprowadzanych badań statystycznych w państwach Unii Europejskiej wynika, że straty wynikłe z niekompetentnych systemów zarządzania, tzn. nieefektywnych metod składania zamówień, lub wielokrotnego wprowadzania tych samych danych (informacji), sięgają średnio 2% zysku[2]. Największą część czasu pracy pracownika biurowego pochłaniają czynności związane z wymianą informacji i komunikacją. Obecnie wiele firm decyduje się na wdrożenie bardzo zaawansowanych systemów, ale i kosztownych. Stosunkowo wysokie koszty wdrożenia zwracają się zwykle po trzech do pięciu lat, oczywiście w przypadku większych firm, a ponoszone ryzyko według doświadczeń amerykańskich, do 60% wdrożeń kończy się niepowodzeniem, ale dotyczy to przede wszystkim specjalizowanych aplikacji pisanych na zamówienie klienta i jest ceną przyszłego sukcesu.

Istota rozproszonej struktury organizacji polega przede wszystkim na zbliżeniu jednostek biznesowych do rynku, oraz przynajmniej częściowej decentralizacji. Systemy wspomagające zarządzanie [1][2][8][7] pomagają przetworzyć informacje o działalności rozproszonej korporacji i budować spójne plany strategiczne, zgodnie z koncepcją "myśl globalnie, działaj lokalnie", zarządzać przede wszystkim informacją.

### **Rozwój i standaryzacja systemów wspomagających zarządzanie**

Przy opracowywaniu profesjonalnych systemów najważniejsze jest opieranie się na standardach, gwarantując w ten sposób współpracę różnych systemów, jak i możliwość dość szybkiego przejścia użytkowników z systemu jednego producenta na inny. Standard MRP (Material Requirements Planning) [1][2][8] czyli metoda Planowania Potrzeb Materiałowych. Metoda MRP bierze swoje początki w późnych latach 50-tych, kiedy to opracowano jej pierwszą wersję - MRP I, w Polsce pierwsze systemy wspomagania zarządzania zaczęły powstawać w latach 70-tych. Metoda MRP I pozwala obliczyć dokładną ilość materia-

łów i terminarz dostaw tak, aby sprostać ciągle zmieniającemu się popytowi na poszczególne produkty, uwzględniając więcej niż jedną fabrykę. W jej nowszych implementacjach bierze się pod uwagę m.in. zamówienia spływające bezpośrednio od końcowych odbiorców (end-user) oraz pośredników, prognozy sprzedaży i produkcji, stany magazynów, zapisy księgowe i fakturowe. Standaryzacji dokonał J.A. Orlicky z firmy IBM w latach siedemdziesiątych.

Model MRP I najlepiej obrazują jego cele, a mianowicie:

- redukcja zapasów;
- dokładne określanie czasów dostaw surowców i półproduktów;
- dokładne wyznaczanie kosztów produkcji;
- lepsze wykorzystanie posiadanej infrastruktury, tzn. magazynów, możliwości wytwórczych;
- szybsze reagowanie na zmiany zachodzące w otoczeniu;
- kontrola poszczególnych etapów produkcji.

Rozszerzeniem specyfikacji MRP I było uwzględnienie Closed Loop MRP (zamknięta pętla sterowania), czyli planowania materiałowych zdolności produkcyjnych w zamkniętej pętli procesu produkcyjnego.

MRP II (Manufacturing Resource Planning - Planowanie Zasobów Produkcyjnych)[1][2][8] to najpowszechniej obecnie stosowany, kompleksowy system planowania procesu produkcyjnego, ułatwiający koordynowanie pracy korporacji, także tej o rozproszonej strukturze. W modelu MRP II większy nacisk w stosunku do poprzedniego został położony na rozbudowanie o elementy związane z procesem sprzedaży i wspierające podejmowanie decyzji na szczeblach strategicznego zarządzania produkcją. W miarę rozwoju, specyfikacja MRP obejmowała kolejne obszary działalności przedsiębiorstwa, stając się stopniowo narzędziem kompleksowym. W modelu MRP II bierze się pod uwagę wszystkie sfery zarządzania przedsiębiorstwem związane z przygotowaniem, planowaniem i kontrolą produkcji, oraz sprzedażą i dystrybucją wyprodukowanych produktów.

W przypadku rozważań dotyczących implementacji modelu MRP II w przemysłowych przedsiębiorstwach produkcyjnych, można wyłonić z opisu standardu następujące funkcje, które powinny być spełnione[1][2]:

- Sterowanie warsztatem produkcyjnym. (*Shop Floor Control SFC*)
- Sterowanie stanowiskiem roboczym. (*Input/Output Control I/OC*)
- Narzędzia i pomoce warsztatowe. (*Tooling Planning and Control*)

Kolejnym krokiem przeistoczenia się metodyki MRP był model ERP (*Enterprise Resource Planning - Planowanie Zasobów na potrzeby Przedsię-*

wzięć)[2], przez wielu zwane po prostu *MRP III (Money Resource Planning - Planowanie Zasobów Finansowych)*.

Specyfikacja ERP wprowadzona była w życie w latach dziewięćdziesiątych, a jej głównym celem jest uzyskanie możliwie jak największej integralności podsystemów infrastruktury informatycznej przedsiębiorstwa na wszystkich szczeblach zarządzania.

System zarządzania klasy ERP powinien integrować następujące podsystemy:

- obsługa klientów - (CRM, EDI).
- produkcja - (MRP I/II).
- finanse - księgowość, raportowanie itp.

W związku z rozwijającym się marketingiem (sprzedażą) pojawiło się nowe zapotrzebowanie na podsystem informatyczny kategorii CRM (Customer Relationship Management) [7], czyli system ukierunkowany na analizę potrzeb klienta (baza informacji o kliencie, jego preferencjach i upodobaniach z kompleksową analizą jego zachowań aby wyjść mu naprzeciw z odpowiednią ofertą). Drugim pojęciem jest EDI (*Electronic Data Interchange*) - *Elektroniczna Wymiana Danych*, jest to standard stanowiący jeden ze składników systemu logistycznego przedsiębiorstwa, wprowadzający bezpapierową ewidencję dokumentów. Technologie wspierające EDI są ciągle rozwijane i ściśle powiązane ze zjawiskiem e-commerce, B2B itp.. W krajach Unii Europejskiej promowany jest standard EDIFACT[2] czyli EDI For Administration, Commerce and Transport. Do realizacji dokumentacji elektronicznej są wykorzystywane formaty danych tekstowych umożliwiające wielokrotne używanie tych samych dokumentów w różnym kontekście i w różny sposób. Dokumenty w postaci elektronicznej mają najczęściej postać hipertekstu, czyli tekstu z dodanym trzecim wymiarem. Ten trzeci wymiar to możliwość wyróżniania fragmentów tekstu lub definiowania go jako odnośnika (link) do innej części dokumentu lub też innych publikacji. Następną ważną możliwością hipertekstu jest wprowadzanie indeksów: tematycznych, chronologicznych, indeksów autorów, osób itp. Indeksy mogą być realizowane w sposób z góry ustalony czyli „statycznie” albo „dynamicznie” generowane przez mechanizmy wyszukiwawcze. Obecnie najbardziej rozpowszechnionym formatem dokumentów elektronicznych jest HTML (*Hypertext Mark-Up Language*)[3]. Większość formatów elektronicznego przechowywania informacji nie nadaje się do sporządzania dokumentacji służącej jako podstawę do różnych form prezentowania czyli mających możliwość automatycznego konwertowania ich do formatu aktualnej prezentacji czy formatu przechowywania. Informacja zawarta w dokumentacji powinna być poddana strukturalizacji co zapewniłoby efektywne i wielokrotne wykorzystanie dokumentacji w wielu mediach zapisu, przekazu lub prezentacji. Do opracowywania doku-

mentacji posiadającej logiczną strukturę jest używany standard SGML (ang. *Standard Generalized Markup Language*) który został opublikowany w 1986 r. jako ISO 8879. SGML służy do tworzenia formalnych definicji struktury dla różnych typów dokumentów. Ciągłe trwają prace nad unowocześnianiem i tworzeniem formatów hipertekstowych. Najnowszą obecnie technologią strukturalnego przechowywania i prezentacji informacji jest XML (ang. eXtensible Markup Language), standard tej technologii nie został jeszcze odpowiednio określony i ciągle się rozwija. Praktycznie XML jest następcą HTML'a, wyposażoną w strukturalną ogólną budowę. Informacja zapisana za pomocą XML jest bardziej ukierunkowana na treść niż na prezentację niż w przypadku standardu HTML. Istnieje wiele formatów elektronicznej dokumentacji, podczas projektowania aplikacji należy zastosować najbardziej standardową specyfikację formatu dokumentów tak, aby zapewnić ich wymianę z oprogramowaniem innych niezależnych producentów.

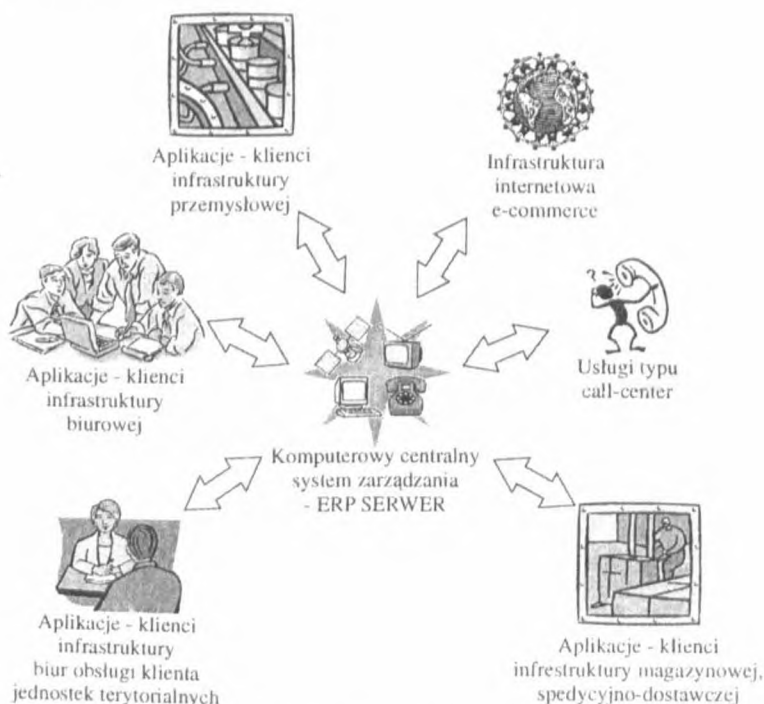
### **Model aplikacji wspomagającej zarządzanie**

Konstrukcję systemu zarządzania trzeba rozpocząć od szczegółowego planu projektowego określającego merytoryczne zastosowanie systemu, zgodnie z czynnikami organizacyjnymi, prawnymi i geograficznymi. W konstrukcjach przewidzianych do seryjnej sprzedaży (tzw. oprogramowanie z półki), przeznaczonych szczególnie dla małych firm, obierany jest ogólny uproszczony model infrastruktury tak aby spełnił on swe podstawowe wymagania. W tworzeniu indywidualnego projektu „na zamówienie” powinny bardzo dynamicznie uczestniczyć osoby szczególnie zainteresowane jego użytkowym charakterem. Bardzo często w przypadku indywidualnych projektów osoby zainteresowane jak najlepszą użytecznością systemu nie mają z góry precyzyjnie określonych wymagań. Często zdarza się, że osoby z poza branży IT są błędnie przekonane że komputer wykona za nie całą pracę, niestety rzeczywistość jest inna i maszyna może pomóc w pracy, ale tylko na tyle i w taki sposób w jaki nauczy ją programista. Aktywna współpraca ze strony przyszłych użytkowników systemu z konstruktorami/programistami doprowadziłaby zapewne do wypracowania odpowiedniego kompromisu, co przyczyniło by się istotnie do sukcesu wdrożenia systemu.

Istnieje wiele trendów związanych z konstruowaniem systemów wspomagających zarządzanie, jednym z takich trendów jest oparcie się wyłącznie na zastosowaniach internetowych, a ściślej związanych z technologiami web'owymi (W3) [14], takimi jak HTML, DHTML, Perl, Java. Takie rozwiązania posiadają wiele zalet (np. niezależność od platformy systemowej), jednak posiadają także wiele wad, cała infrastruktura sieciowa jest bardziej obciążona

poprzez przesyłanie nie tylko istotnych danych, lecz także szeregu informacji związanych z wizualizacją tych danych. Rozwiązania takie wymagają ciągłego połączenia z jednostką centralną, brak rozwiązań lokalnego buforowania informacji może mieć konsekwencje związane z przestojem działalności w przypadku awarii medium transmisji. Nadmiarowość danych nie jest wskazana ze względów na zwiększony koszt implementacji oraz utrzymania medium transmisji, szczególnie wtedy gdy jednostka terenowa znajduje się w odległym od dużych aglomeracji miejscu np. stacje benzynowe. Nawet w przypadku lokalnej infrastruktury sieciowej, nisko obciążone łącza mogą posłużyć do zaimplementowania innych usług wspierających ogólnie pojęte zarządzanie np. usług telekomunikacyjnych, video konferencji, systemów kontroli dostępu i zabezpieczeń, video-monitoring.

W dalszej części rozważane będą konstrukcje aplikacyjne związane z szeroko pojętym zarządzaniem klasy ERP z naciskiem na wsparcie zarządzania produkcją (MRP).



Rys. 1 Uogólniony schemat systemu zarządzania.

Systemy klasy ERP (Rys. 1) oprócz wspierania biurokracji przedsiębiorstwa powinny w pełni integrować strategiczne podsystemy automatyki przemysłowej, poprzez komunikację z aplikacjami nadzorującymi etapy produkcji lub też bezpośrednio z samymi urządzeniami sterującymi tą produkcją, wykorzystując standardy przemysłowe (np. komunikacja szeregową RS-485, protokół komunikacyjny MODBUS). Jako przykład można sobie wyobrazić automatyczną linię pakowania produktów, zintegrowaną z systemem zarządzania, takie rozwiązanie da natychmiastową informację o ilości produktów gotowych do sprzedaży.

Aby jak najbardziej efektywnie zintegrować urządzenia przemysłowe z systemem zarządzającym należy wykonać odpowiednie aplikacje pracujące w środowisku sieciowym<sup>1</sup> [5]. Oprogramowanie wspomagające zarządzaniem przedsiębiorstwa w dzisiejszych czasach nie może ograniczyć się fizycznie tylko do jednego komputera, zatem takie oprogramowanie musi być aplikacją rozproszoną<sup>2</sup> [5].

Istotą koncepcji aplikacji rozproszonej jest to, aby środowisko przetwarzania danych z punktu widzenia użytkownika wyglądało jak przetwarzanie na lokalnej maszynie, a topografia rozmieszczenia komputerów w sieci była niewidoczna. W praktyce operator takiego oprogramowania nie ma możliwości stwierdzenia gdzie i jak są przetwarzane lub przechowywane dane, czy to na lokalnym komputerze czy też na komputerze odległym (zdalnej maszynie). Aby aplikacje rozproszone mogły ze sobą współpracować konieczne jest określenie mechanizmu komunikacji. W większości dużych, średnich i małych firm (przedsiębiorstw), w ich wewnętrznych sieciach komputerowych zastosowany jest protokół<sup>3</sup> TCP/IP<sup>4</sup> [5] spełniający mechanizm komunikacji. Tworzone aplikacje rozproszone powinny zatem wykorzystywać usługę transportową protokołu TCP/IP. Programiści wykorzystujący ten protokół w swoich aplikacjach nie muszą znać szczegółów budowy tego protokołu, szczególnie ważna jest termi-

---

<sup>1</sup> Sieć (network) nazywamy zestaw komputerów połączonych ze sobą mogących wzajemnie się komunikować, współdzielić korzystanie z urządzeń zewnętrznych, a także mieć dostęp do innych sieci.

<sup>2</sup> Aplikacją rozproszoną nazywamy taki zespół programów działających na różnych komputerach, współpracujących ze sobą poprzez sieć i realizujących określone zadanie przetwarzania danych.

<sup>3</sup> Protokołem nazywamy zbiór zasad, reguł, służących do wymiany danych (informacji) pomiędzy dwoma maszynami lub procesami - czyli programami dynamicznie wykonywanymi przez system operacyjny.

<sup>4</sup> TCP/IP (ang. Transport Control Protocol/Internet Protocol) jest to taki protokół komunikacyjny definiujący dolne warstwy sposobu komunikacji programów sieciowych, szczególnie internetowych.

nologia z nim związana i nisko poziomowy protokół adresowania IP. Aby aplikacje mogły ze sobą współdziałać, programista musi określić interakcję po między programami tworzącymi aplikację rozproszoną. Przebieg interakcji między programami współpracującymi na warstwie transportowej wybranego protokołu komunikacyjnego (np. TCP/IP) nazywamy protokołem aplikacyjnym (application protocol). Można wyróżnić protokoły aplikacyjne standardowe i niestandardowe. W praktyce za każdym razem, gdy programista pisze nową aplikację korzystając z protokołu komunikacyjnego, tworzy nowy protokół aplikacyjny. Protokoły aplikacyjne udokumentowane w dokumentach RFC stają się standardowymi protokołami aplikacyjnymi [5] (standard application protocol), a wszystkie inne są niestandardowymi protokołami aplikacyjnymi (nonstandard application protocol). Ze standardowych protokołów aplikacyjnych można wyróżnić np.: przesyłanie plików (file transfer), wysyłanie i odczytywanie poczty elektronicznej (electronic mail), pracy na odległym komputerze, czyli usługa telnet. Mechanizm transportowy protokołu komunikacyjnego TCP/IP umożliwia programiście jedynie nawiązanie komunikacji i przesłanie danych pomiędzy dwoma aplikacjami działającymi na lokalnym lub odległym komputerze, zatem zapewnia on komunikację pomiędzy równo uprawnionymi partnerami zwaną partner-do-partnera (peer to peer communication). Programista musi określić zasady prowadzenia interakcji, czyli zbudować protokół aplikacyjny. W praktyce najczęściej stosuje się model organizacji aplikacji rozproszonych, korzystających z protokołu TCP/IP zwany jako model klient-serwer (client-server paradigm) [5]. Model ten stał się podstawowym wzorem określania interakcji między aplikacjami korzystającymi z komunikacji sieciowej partner do partnera. Podstawową potrzebą powstania modelu klient-serwer było wyeliminowanie problemu nawiązywania połączenia pomiędzy aplikacjami, zwanego problemem spotkań (problem of rendezvous). W protokole TCP/IP nie ma wbudowanych mechanizmów automatyzacji inicjowania połączenia, czyli aby program mógł odebrać informację musi na nią czekać. W modelu klient-serwer rozróżniamy dwie kategorie programów, program kategorii klient, który nawiązuje połączenia (generuje zgłoszenie, żądanie wykonania usługi), oraz program kategorii serwer, który ciągle oczekuje na nawiązanie połączenia generowanego przez program klienta dając mu odpowiedź. Program serwera jest znacznie trudniejszy do napisania w stosunku do programu klienta. Serwer zazwyczaj musi obsłużyć większą liczbę programów typu klient zatem musi mieć zagwarantowaną uprzywilejowaną pracę w systemie dla zapewnienia optymalnej wydajności. Wiąże się to z zaimplementowaniem w programie serwera kontroli dostępu do danych i weryfikowania uprawnień użytkowników pracujących na programach typu klient, ponieważ serwer pracuje w systemie operacyjnym jako program uprzywilejowany to brak takich ograniczeń może doprowadzić do dziur w ochronie systemo-



wego serwera na którym wykonywać się będzie nasza aplikacja. Interakcja w aplikacji typu klient-serwer bazującej na protokole komunikacyjnym TCP/IP jest połączeniowa (connection-oriented), zatem jest w pełni niezawodna zarówno w sieci lokalnej i rozległej. Programista korzystając z technologii TCP nie musi martwić się o przebieg transmisji, kontrola dotarcia pakietów danych, ich kolejności, oraz szybkość przepływu danych wykonywana jest automatycznie. Kod protokołu TCP zaimplementowany w systemie operacyjnym na, którym jest wykonywana aplikacja sprawdza, czy każdy segment danych dotarł do odbiorcy, jeżeli nie to automatycznie wykonywana jest ponowna transmisja. Na podstawie sumy kontrolnej przesyłanych danych zachodzi kontrola wiarygodności danych. W celu wyeliminowania duplikacji segmentów danych, oraz ich niewłaściwej kolejności, każdy segment zostaje oznaczony numerem porządkowym. W przypadku niemożności podjęcia transmisji zarówno część aplikacji rozproszonej kategorii serwer jak i klient zostaje automatycznie powiadomiona. W przypadku projektowania aplikacji wspomagających zarządzanie, lub w każdym innym przypadku aplikacji rozproszonej, gdy nie można pozwolić sobie na utratę danych należy odpowiednio zaprojektować interakcję opartą o protokół komunikacyjny TCP. Innym ważnym kryterium świadczącym na korzyść protokołu TCP jest fakt mniejszego zaangażowania pracy w tworzoną aplikację, zmniejszeniu liczby programistów, skróceniu czasu opracowywania i implementacji protokołu aplikacyjnego, określającego interakcję w budowanej aplikacji rozproszonej. Dla zapewnienia skalowalności tworzonej aplikacji należy uwzględnić możliwość zmian numeru IP [5] oraz wykorzystywanego do nawiązania połączenia numeru portu, zarówno w programie serwera jak i w programie klienta. Aplikacje umożliwiające wprowadzanie strategicznych zmian w funkcjonalności technicznej noszą nazwę programów w pełni sparametryzowanych (fully parameterized) [6].

Aby wiele programów typu klient pracowało ciągle z serwerem, czyli z punktu widzenia użytkowników ich programy typu klient przetwarzały dane jak aplikacja lokalna w tym samym czasie, należy rozważyć i zdefiniować pojęcie serwera współbieżnego. Podstawową jednostką zasobów i przebiegu przetwarzania danych w programach współbieżnych jest proces lub inaczej wątek<sup>5</sup>. W systemie o przetwarzaniu współbieżnym wiele wątków, może wykonywać tą samą, lub inną część kodu programu, każdy z wątków może rozpocząć się, wstrzymać działanie lub zakończyć się w dowolnej chwili i każdy z nich może przetwarzać dane z dowolną określoną przez programistę szybkością. Oczywiście równoległa praca wielu wątków, jest iluzją systemu operacyjnego, który każdemu z wątków przydziela cyklicznie z bardzo dużą szybkością krótki czas

---

<sup>5</sup> Wątek (ang. thread). jest to praca części kodu programu niezależna od działania samego programu.

aktywacji. Z punktu widzenia człowieka wątki działają jednocześnie. Istnieją także komputery wieloprocesorowe, które dzielą sobie sprzętowo wykonywanie wątków pomiędzy procesory.

Serwer systemu wspomagającego zarządzanie musi bezpiecznie przechowywać dane oraz w efektywny sposób je udostępniać do edycji lub analiz. Wszystkie dane są przechowywane w bazie danych<sup>6</sup> którą z powodzeniem można nazwać mózgiem systemu wspomagającego zarządzanie.

Technologia przechowywania danych rozwija się praktycznie od samego początku komputeryzacji. Przetwarzanie informacji nie może odbywać się bez przechowywania danych wejściowych czy też rezultatów pewnych operacji. Obecnie najefektywniejsze są relacyjne (RDBMS<sup>7</sup>) lub relacyjno-obiektowe bazy danych, w których dane przechowywane są w postaci rekordów, które wraz z odpowiednimi kluczami, czyli indeksacją są połączone relacjami, odzwierciedlającymi odpowiednią strukturę danych reprezentującą potrzebne informacje systemu zarządzania. Poprzednie technologie oparte o proceduralne schematy przetwarzania danych w bazie pracowały niezawodnie, gdy operacje na danych miały wyraźny logiczny początek i koniec. Systemy wspomagające zarządzanie, są środowiskami transakcyjnymi, gdzie przetwarzanie danych odbywa się, za pomocą licznych zbiorów rekordów. Praca na pojedynczym rekordzie jest nieoptymalna, a wręcz kłopotliwa, dlatego należy stosować nieproceduralne schematy przetwarzania danych.

System RDBMS spełnia następujące zadania :

- Utrzymuje dane w określonej postaci, za pomocą procedur niskiego poziomu.
- Przeprowadza wszelkie operacje we/wy związane z plikami, lub urządzeniami, będącymi zasobami systemu operacyjnego komputera, na którym działa.
- Zapewnia mechanizmy dostępu do danych.
- Redukuje wszelkie anomalie w przechowywaniu danych.
- Optymalizuje sposób przechowywania danych .
- Zapewnia spójność i integralność przechowywanych informacji.

---

<sup>6</sup> Baza danych (database) jest to program przechowujący w określony sposób dane, umożliwiający ich udostępnianie w postaci rzeczywistej, lub przetworzonej. Programy tego typu możemy określać terminem „serwera danych”, lub „hurtownią danych”.

<sup>7</sup> RDBMS (ang. Relational Database Management Systems)[5] jest relacyjny system zarządzania bazami danych, zwany potocznie serwerem baz danych lub hurtownią danych.

Projekt logiczny powinien być opracowywany wraz z osobami zainteresowanymi powstającym oprogramowaniem, najlepiej nadawaliby się eksperci z dziedziny w jakiej ma być używana baza danych. Ekspert powinien określić sensowny z uwzględnieniem dalszego rozwoju zestaw tabel i relacji, na podstawie którego programista (projektant aplikacji), mógł go według określonych technologii zakodować. Zespół ekspertów biorących udział w projektowaniu aplikacji wspomagającej zarządzanie, powinien mieć praktyczną wiedzę z następujących dziedzin [7]:

- Rachunkowość i finanse przedsiębiorstwa (pełna księgowość)
- Logistyki przedsiębiorstwa.
- Zarządzania finansami.
- Zarządzania inwestycjami.
- Zarządzania zasobami ludzkimi.
- Zarządzania produkcją.
- Handlu i obrotu towarami wewnętrznego i międzynarodowego.
- Prawa gospodarczego, podatkowe, celnego i praw pośrednich.
- Zarządzania warsztatem produkcyjnym (technolog produkcji)

Współpraca zespołu ekonomistów z prawnikami jest niezbędna, aby powstała aplikacja była w pełni użyteczna, a przede wszystkim zgodna z powszechnie obowiązującymi przepisami prawa. Cała ogólna konstrukcja aplikacji powinna łączyć analitykę, a także kreatywność i intuicję. Model bazy powinien posiadać niezbędną ilość tabel nadmiarowość jest niekorzystna, przy jednoczesnym zachowaniu elastyczności, otwartości systemu w przypadku późniejszych uaktualnień, czy też ewentualnego rozwoju. Eksperci będący źródłem projektu logicznego, nie powinni być zdziwieni rozwojem ich dziedziny w czasie żywotności aplikacji, pomijając oczywiście rewolucyjne zmiany. Ogólnie wiadomo, że biurokracja jest zbudowana poprzez szereg komplikacji, mnogość wszelkiego typu dokumentów i formularzy znacznie utrudnia zaprojektowanie systemu wspomagającego zarządzanie. W projektowaniu odpowiedniej bazy danych komplikacje wynikają z [7]:

- Słabego projektu logicznego, bez odpowiednio sformułowanych atrybutów i priorytetów.
- Powiązań z innymi, już istniejącymi nieprawidłowymi systemami.
- Rozwiązań budowanych na ograniczonych zasobach.
- Niemożności zredukowania złożonej semantyki.

Zbyt uogólniony model danych spełniający próbę zadowolenia wszystkich użytkowników, da w rezultacie model całkowicie znormalizowany i czysty, ale tak potężny, że tylko niewielka liczba użytkowników jest w stanie go zrozumieć.

Pewnym rozwiązaniem komplikacji są zasady normalizacji danych dostarczające odpowiednie testy dla modelu i określane terminem CASE<sup>8</sup> [4]:

Całkowicie znormalizowany projekt pozwala zapobiegać większości komplikacji zwanych z anomaliami, które mogłyby zagrażać przetwarzaniu i jakości przechowywanych w bazie informacji. Aby baza danych była znormalizowana, trzeba rozważyć, czy spełnia ona kilka warunków zwanych postaciami:

*Pierwsza postać normalna* określa brak powtarzających się grup informacji. Dane powinny być wartościami elementarnymi nierozkładalnymi. Oczywiście określenie wartości elementarnej jest rzeczą umowną, charakterystyczną dla danego zagadnienia. Podstawowym problemem baz danych jest redundancja, czyli nadmiarowość, powtarzalność danych. W przypadku aplikacji wspomagających zarządzanie musimy rozważyć podchodzić do redundancji, gdyż w wielu przypadkach musimy mieć powtarzające się informacje, co wiąże się z wymogami ekonomicznymi i wymogami powszechnie obowiązującego prawa. Przykładem mogą być wystawione rachunki sprzedaży, kiedy to kontrahent zmieni choćby raz adres siedziby firmy w tym samym roku bilansowym. Wprowadzenie zmian adresu firmy w tym samym rekordzie spowodowałoby, że wszystkie rachunki uległyby zmianie. Zachodzi zatem potrzeba redundancji danych dla zachowania ich spójności w czasie, jest to szczególnie ważne, gdyż wszelkiego rodzaju dokumenty muszą być przechowywane przez okres pięciu, a nawet dziesięciu lat.

Niespójność informacji w bazie danych określamy jako anomalie [4]:

Anomalie przy aktualizacji, anomalie przy wprowadzaniu danych (np. w momencie gdy nie można wprowadzić danych klienta, który nic jeszcze nie kupił.), anomalie przy usuwaniu danych (np. w momencie anulowania transakcji, może wystąpić usunięcie informacji o kliencie.)

*Druga postać normalna* określa zależność wszystkich atrybutów, od jednoznacznego identyfikatora. Postać ta stosuje się wyłącznie do tabel z unikalnym kluczem złożonym. Należy bardzo uważnie rozważyć znaczenie atrybutów, gdyż nie istnieją żadne metody pozwalające na automatyczne wyznaczenie za-

---

<sup>8</sup> CASE\*Method Entity Relationship Modeling, czyli zasady modelowania związków encji. Metoda ta została opracowana przez Richarda Barkera w 1989 r., procedura wymusza pewne użyteczne standardy modelowania danych, które zapobiegają nadmiarowi zbytecznych danych i pozwalają na utożsamienie modelu bazy z projektem fizycznym, co sprowadza się do całkowitego znormalizowania projektu logicznego.

leżności, lecz wynikają one z wiedzy projektanta (eksperta) o naturze i zachowaniu danych w bazie.

*Trzecia postać normalna* formułuje twierdzenie że żadna wartość atrybutu nie będącego częścią klucza, nie zależy od innego atrybutu, nie będącego częścią klucza. Postać ta uniemożliwia określenie przez atrybuty swoich wartości i jest najważniejszą z postaci. Odmianą trzeciej postaci normalnej jest postać normalna Boyce'a-Codda, która jest najsilniejsza ze wszystkich znanych postaci normalnych dla zależności nie wielowartościowych.

Relację która nie spełnia zasady postaci Boyce'a-Codda, można przekształcić do tej postaci poprzez rozłożenie schematu relacji. Ogólnie przyjęte jest, aby relacje spełniały przynajmniej trzecia postać normalna, albo najlepiej postać normalną Boyce'a-Codda, która nie zawiera redundancji, wszelkich anomalii oraz niezgodności.

Podczas wyboru bazy danych do implementacji w systemie ważne jest aby w pełni wspierała technologię SQL<sup>9</sup>, zapewniając efektywniejsze żonglowanie danymi i przyspieszenie czas implementacji, co wiąże się ze zmniejszeniem kosztów. Na rynku istnieje wiele technologii baz danych, wybór wiąże się z charakterystyką przedsięwzięcia i oczywiście z kosztami wykupienia licencji. W przypadku dużych przedsiębiorstw z powodzeniem można implementować kosztowne produkty takie jak IBM DB2 [13], Oracle [11], Microsoft SQL Server [12]. W przypadku większego nie obliczania się z kosztami wybór jest dość prosty, wszystkie produkty zapewniają wysokiej jakości technologię przetwarzania danych, są niezależne od platformy( z wyjątkiem MS SQL Serwer), oraz są opakowane szerokim wachlarzem funkcji ułatwiających dość szybkie wdrożenie. Problem rozpoczyna się w przypadku „taniach” rozwiązań, przewidzianych dla małej lub średniej wielkości przedsiębiorstw, szczególnie w przypadku systemów przewidzianych do seryjnej sprzedaży. I tu także można znaleźć ciekawe rozwiązania, stosując na przykład technologię CODEBASE [10] tanią i efektywną, lub zastosować system bazy danych MySQL [9]. System ten jest bardzo ciekawą pozycją, pomimo swej surowej prostoty w niektórych zastosowaniach może konkurować nawet z systemem firmy Oracle. Posiada dość spore możliwości jak na cenę licencji nie przekraczającą 300\$ USD, poniżej przedstawione są przykładowe możliwości:

- Dostępny na wielu platformach (Windows, Unix, Linux, FreeBSD, itp.).
- Zapewniona praca wielowątkowa (multi-user).

---

<sup>9</sup> SQL (ang. Structured Query Language)[5] jest to strukturalny język zapytań zawierający standardowy zestaw poleceń służących do komunikowania się z większością najnowszych technologii baz danych RDBMS, dostępnych na rynku.

- Posiada wsparcie rozwojowe dla wielu języków programowania (C, C++, Java, Perl, PHP, itp.).
- W pełni zgodny ze standardem ANSI SQL (SQL'97).
- Wspiera komunikację TCP/IP.
- 8-bajtowa reprezentacja liczb całkowitych.
- Sprawdzone działanie w konfiguracji 60,000 tabel i 5,000,000,000 wierszy.

System bazy danych MySQL [8] został pomyślnie sprawdzony w wielu aplikacjach Internetowych (portale), zatem powinien sprawdzić się także w aplikacjach wspomagających zarządzanie związanymi, także z e-commerce, przewidzianych dla nisko budżetowych klientów.

Zakończywszy powyższe wstępne rozważania dotyczące aplikacji serwera przejdźmy do aplikacji typu klient. Z wcześniejszych rozważań wiemy że aplikacja klienta w celu uzyskania dostępu do danych, musi nawiązywać połączenie z aplikacją serwera, są to bardzo ważne aspekty technologiczne, jednak jest jeszcze istotniejsza sprawa projektowania aplikacji, dotycząca interfejsu użytkownika. W przypadku aplikacji serwera którego użytkowanie należy powierzyć specjalistom z branży IT, to w przypadku aplikacji typu klient, którą będą obsługiwali zwykli użytkownicy (operatorzy komputera) wypadało by użyć graficznego interfejsu użytkownika (GUI<sup>10</sup>) [6]. Zadowolenie końcowych użytkowników, nie mający często praktycznie żadnej wiedzy informatycznej, jest miarą przyszłego sukcesu konstruowanego systemu. Użytkownicy nie docenią projektanta aplikacji za skomplikowane algorytmy przetwarzania danych oraz ciężką i mozolną ich implementację. Nie widząc zachodzących procesów, będą osądzać aplikacje po jej wyglądzie zewnętrznym i sposobie użytkowania, zatem należy poświęcić dość dużo czasu na opracowanie interfejsu użytkownika, aby aplikacja została przyjęta przez większość użytkowników z entuzjazmem, tylko większość, gdyż wszystkich nie da się zadowolić. Jeżeli chodzi o wybór platformy operacyjnej dla projektowanej aplikacji klienta, należy zastosować platformę rodziny Windows firmy Microsoft [11]. Podczas tworzenia aplikacji podstawowym jej aspektem jest funkcjonalność i użyteczność dla przyszłego użytkownika, dlatego trzeba poczynić wiele starań aby praca z programem była przyjemna lub choćby bezstresowa. System Windows zdobył sobie ogromną popularność właśnie dzięki przejrzystemu i mało skomplikowanemu interfejsowi, poprzez który użytkownik komunikuje się z komputerem. Jednym z głównych aksjomatów projektowania oprogramowania jest słynna zasada minimalizacji szoku SMP

---

<sup>10</sup>GUI ( Graphical User Interface) Metoda komunikacji pomiędzy użytkownikiem a komputerem realizowane, za pomocą intuicyjnych graficznych symboli, nazywana graficznym interfejsem użytkownika.

(Shock Minimalization Principle). Według zasady SMP oprogramowanie powinno działać tak, aby minimalizować szok wywierany na użytkownika. Przedsiębiorstwo z tak zbudowanego system też odniesie korzyści wynikające z mniejszej potrzeby doskonalenia kadry pracowniczej, systemy i produkty firmy Microsoft są powszechnie stosowane na różnorodnych kursach i szkoleniach, co znacznie obniża koszty wdrożenia. Istnieje wiele środowisk programistycznych dla systemów operacyjnych rodziny Windows jednak wskazane jest użycie Microsoft Visual C++ 6.0 lub Microsoft Visual C# [12] dla nadchodzącej platformy Windows .Net. Technologia programowania obiektowego C++ lub C# daje dość proste mechanizmy łączenia funkcji przetwarzania danych z interfejsem użytkownika, oraz transmisji danych, poprzez sieć komputerową. Platforma Windows daje duże możliwości wizualizacji danych poprzez możliwość osadzania dokumentów innych aplikacji istnieje możliwość integracji projektowanego systemu z innymi programami związanymi z edycją dokumentów (np. MS Office) lub aplikacjami wspomagającymi projektowanie (np. AutoCAD) co znacznie usprawni proces obiegu dokumentacji i podniesie walory użytkowe projektowanego systemu, tak aby spełnić oczekiwania klient. Autonomiczne aplikacje pracujące pod kontrolą systemu Windows mogą też łatwo współpracować z urządzeniami zewnętrznymi takimi jak drukarki fiskalne, czytniki kodów kreskowych, oraz z urządzeniami stosowanymi w automatyce przemysłowej.

## Wnioski

Projektowanie aplikacji wspomagającej zarządzanie przedsiębiorstwem przemysłowym jest dość skomplikowane, wymaga od konstruktorów ogromnej wiedzy praktycznej. Złożoność integracji podsystemów ekonomicznych i przemysłowych nastęrcza wiele problemów i wymaga wielu kompromisów.

W związku z stosowaniem różnego rodzaju technologii produkcyjnych, każde przedsiębiorstwo jest inne i nie możliwe jest sprecyzowanie jednolitego modelu aplikacji wspomagającej zarządzanie, a ciągły postęp technologiczny wywiera nacisk na otwartość systemu, w celu późniejszych modyfikacji, szczególnie infrastruktury przemysłowej.

## Źródła

1. Darryl V. Landvater, Christopher D. Gray „*MRP II Standard System*”; Oliver Wight Publications, 1989
2. M. Janiec „*Komputerowe systemy wspomagania zarządzania klasy MRP/ERP*”; Akademia Ekonomiczna w Krakowie, 1997.
3. MSDN Library Visual Studio 6.0+ „*Microsoft Windows User Experience*” dokumentacja techniczna.
4. Hans Ladanyi „*SQL Księga eksperta*”; Wydawnictwo HELION, 2000
5. Douglas E. Comer, David L. Stevens „*Internetworking with TCP/IP*”; Department of Computer Sciences Purdue University, West Lafayette, IN 47907
6. John Robbins „*Debugging Applications*”; Microsoft Press, 2001
7. R. Ambroziak „*Tworzenie oprogramowania wspomagającego zarządzanie przedsiębiorstwem przy użyciu pakietu MS Visual C++ 6.0*”; Katedra Informatyki Stosowanej PŁ, 2000, dyplomowa praca magisterska.
8. Adamczewski J., *Zintegrowany system informatycznego zarządzania*, „Logistyka”, nr 4, 1997.
9. [www.mysql.com](http://www.mysql.com)
10. [www.codebase.com](http://www.codebase.com)
11. [www.oracle.com](http://www.oracle.com)
12. [www.microsoft.com](http://www.microsoft.com)
13. [www.ibm.com](http://www.ibm.com)
14. [www.w3.org](http://www.w3.org)