

**Polska Akademia Nauk
Instytut Badań Systemowych**

Ryszard Budziński

SYSTEM RACHUNKOWOŚCI TRANSAKCYJNEJ

- algorytmy
- metodyka
- informatyka

Warszawa 1998

*matice
vycepani*



SYSTEM RACHUNKOWOŚCI TRANSAKCYJNEJ

Polska Akademia Nauk · Instytut Badań Systemowych

Seria: BADANIA SYSTEMOWE
tom 22

Redaktor naukowy:
Prof. dr hab. Jakub Gutenbaum

Warszawa 1998

Ryszard BUDZIŃSKI

SYSTEM RACHUNKOWOŚCI TRANSAKCYJNEJ

- algorytmy
- metodyka
- informatyka

Publikację opiniowali do druku:

Prof. dr hab. Jerzy Kisielnicki

Prof. dr hab. Zenon Głodek

Dyskutowany problem baz temporalnych pojawił się w badaniach autora – w programie badań podstawowych IBS PAN (zlecenie A1630/91). Podstawy metodyczne systemu TRANS autor opublikował również w Zeszytach Teoretycznych Stowarzyszenia Księgowych w Polsce w numerze 19/1992. Najważniejszym momentem wdrożonego do praktyki systemu (w wielu przedsiębiorstwach i w służbie zdrowia woj. szczecińskiego) jest pełne zabezpieczenie potrzeb informacyjnych użytkownika.

Copyright © by Instytut Badań Systemowych PAN
Warszawa 1998

ISBN 83-85847-23-5

ISSN 0208-8029



WYDAWNICTWO I DRUKARNIA
INSTYTUTU INFORMATYKI POLITECHNIKI SZCZECIŃSKIEJ
ul. Żołnierska 49, 71-210 Szczecin, tel. (091) 764 48 56

Nakład 100+24. Ark. druk. 16,5
Grudzień 1998 r.

SPIS TREŚCI

1. WSTĘP	9
2. ZARZĄDZANIE W GOSPODARCE RYNKOWEJ	13
3. EWIDENCJA ZDARZEŃ GOSPODARCZYCH	32
3.1. Problemy reprezentacji czasu	32
3.2. Metody opisu zdarzeń gospodarczych	36
3.3. Systemy informatyczne w rachunkowości	43
3.3.1. Systemy klasyczne	44
3.3.2. Systemy zintegrowane	47
3.3.3. Systemy klient/serwer	49
4. ARCHITEKTURA SYSTEMU TRANSAKCYJNEGO	53
4.1. Założenia metodyczne	54
4.2. Algorytm systemu	59
4.3. Strona tytułowa	66
4.3.1. Użytkownicy systemu	70
4.3.2. Opcje administratora	72
4.3.3. Opcje uzupełniające	85
5. PRACA W SYSTEMIE INFORMATYCZNYM	87
5.1. Praca na transakcjach	89
5.1.1. Księgowanie dowodów	92
5.1.2. Import transakcji	101
5.1.3. Odłączenia transakcji	103
5.1.4. Operacje na schowku	105
5.1.5. Rejestr VAT	106
5.1.6. Dziennik rachunkowości	108
5.1.7. Kontrolka księgowania	109
5.1.8. Eksport transakcji	110
5.2. Reguły automatycznego księgowania	112
5.2.1. Wprowadzanie reguł	114
5.2.2. Korekty reguł księgowania	119
5.2.3. Usuwanie reguł	120
5.2.4. Zamykanie kont	121
5.2.5. Korespondencja reguł	123

5.3. Operacje na kontach	124
5.3.1. Bilans otwarcia zadania	126
5.3.2. Korekty bilansu otwarcia	133
5.3.3. Modelowanie stanu kont	134
5.3.4. Edycje zobowiązań i należności	137
5.3.5. Potwierdzenia księgowania	138
5.3.6. Bilans próbny kont	141
5.4. Korespondencja finansowa	145
5.4.1. Wystawianie rachunków	145
5.4.2. Realizacja przelewów	149
5.4.3. Edytor tekstu	152
6. OPCJE WSPÓLNE DLA ZADANIA	158
6.1. Archiwowanie zadań	158
6.2. Rekonstrukcje danych	159
6.3. Nastawienia i sterowanie	161
6.3.1. Powoływanie tytułów	163
6.3.2. Zmiany atrybutów kont	165
6.3.3. Korekta modelu ZPK	166
6.3.4. Baza adresowa systemu	168
6.3.5. Rygory zobowiązań	170
6.3.6. Nagłówki do korespondencji	171
6.3.7. Kolorystyka ekranu	172
6.3.8. Deklaracja kont VAT	172
6.3.9. Blokada baz i zadania	175
6.3.10. Programowanie drukarek	176
6.4. Zamykanie roku obrachunkowego	177
7. JĘZYK ZAPYTAŃ W SYSTEMIE	180
7.1. Opcje dostępu do danych	180
7.1.1. Kryteria przedziału czasu	181
7.1.2. Filtry podprowadzające	182
7.1.3. Podglądy zbiorów głównych	184
7.1.4. Kryteria pozycji	189
7.1.5. Sortowanie plików	191
7.1.6. Kwalifikacja indywidualna	193
7.2. Edytory przeglądania	194

7.2.1. Przeglądanie kartotek	194
7.2.2. Edycja tekstu	197
7.2.3. Edytor informowania	198
7.3. Projektowanie w systemie	207
7.3.1. Język użytkownika systemu	207
7.3.2. Zarządzanie projektami	221
8. INTEGRACJA W SYSTEMIE RACHUNKOWOŚCI	225
8.1. Integracja dziedzinowa	227
8.1.1. Model systemu informacyjnego	228
8.1.2. Jednostka ewidencji zdarzeń	229
8.1.3. Instytucja zlecenia	232
8.1.4. Reprezentacja czasu	233
8.2. Model obiektowy rachunkowości	235
8.2.1. Podstawowe pojęcia	236
8.2.2. Model konceptualny bazy	237
8.2.3. Organizacja klas danych	240
8.3. Organizacja systemu informatycznego	246
9. ZAKOŃCZENIE	254
PRZYPISY	256
LITERATURA	260

8. INTEGRACJA W SYSTEMIE RACHUNKOWOŚCI

Integracja w rachunkowości jest problemem niezmiernie trudnym, a jednak nieuniknionym z punktu widzenia teorii sterowania. Zarządza się bowiem ogółem i szczegółem równocześnie. Pojęcie integracji jest tu wielopłaszczyznowe. Można ten problem rozpatrywać na tle wielu uwarunkowań, do których w pierwszej kolejności zaliczyć trzeba samą postać systemu informacyjnego, następnie możliwość opisu systemu w strukturach danych, wreszcie sam język zapytań, za pomocą którego komunikujemy się z systemem. Szerszą płaszczyzną tej integracji jest połączenie uzyskanych rozwiązań bazy faktów z metodami identyfikacji (i analizy postępu w efektywności) w kierunku generowania wielokryterialnych rozwiązań decyzyjnych. Proces integracyjny jest, w tym przypadku nieuchronny chociażby z tego punktu widzenia, że współczesne systemy informatyczne przestały być tylko rejestratorami zdarzeń. Następuje ich ewolucja w kierunku pełnego wyłączenia i automatyzacji w pracach biurowych oraz bardziej złożonych pracach wykonawczych, w niedalekiej przyszłości także kierowniczych, poprzez komputerowe wspomaganie decyzji. Głównym stymulatorem tych zamierzeń jest informatyka oraz zmiana nastawienia do postrzegania i opisu rzeczywistości. Dotyczy to technik obiektowych, które coraz mocniej akcentują swoje miejsce w systemach informatycznych zarządzania.

Informatyka jest taką dyscypliną naukową, która zajmuje się badaniem dla potrzeb poznawczych całej nauki, praw rządzących kodowaniem, zapisywaniem, przetwarzaniem i przesyłaniem informacji oraz wykorzystywaniem tych praw dla tworzenia metod i systemów, wedle ludzkich potrzeb.

Cechy charakterystyczne współczesnych kierunków rozwojowych informatyki:

- w zakresie algorytmów i architektury komputerów coraz większą uwagę zwraca się na równoległość wykonywanych operacji — algorytmy równoległe, komputery równoległe,
- w zakresie indywidualnych komputerowych stanowisk pracy coraz większą uwagę zwraca się na łączenie tych stanowisk w zbiory komputerów połączonych odpowiednimi liniami komunikacyjnymi — wysokowydajne sieci komputerowe,
- w coraz większym stopniu zwraca się uwagę na multimedialność postaci informacji — w zakresie ich przetwarzania stosuje się operacje matematyczne, a także operacje takie jak: identyfikacja obrazów, wydzielenie konturów, filtracja, kompresja binarnych zapisów obrazów i głosu,

- następuje rozszerzenie tradycyjnej bazy technologicznej — w zakresie linii komunikacyjnych o technologie optyczne, w zakresie pamięci zewnętrznych o takie rozwiązania, jak np. dyski laserowe, pracuje się nad komputerami multimedialnymi i optycznymi,
- w zakresie programowania przechodzi się na prace w wieloosobowych zespołach, w oparciu o profesjonalne technologie oprogramowania,
- odnośnie algorytmów, które są podstawą programów, dokonywane są prace nad osłabieniem założeń o konieczności absolutnego determinizmu danych.

Tyle o wizji współczesnej informatyki mówi definicja, podawana przez świat nauki i technologie światowe. Już nie są to prognozy, ale rzeczywistość, która w coraz szybszym tempie przenika do codzienności⁷⁰.

Rozwojowi informatyki towarzyszy zmiana nastawienia do sposobu gromadzenia i przetwarzania danych w komputerze. Klasyczne podejście proceduralne i bazy relacyjne w zasadzie wyczerpały swoje możliwości rozwojowe⁷¹. Nowym podejściem jest technika obiektowa, która znalazła zastosowanie w niniejszej pracy w podejściu transakcyjnym do budowy systemu rachunkowości. Teoria zdarzeń (encje, cząstki) nieodłącznie wiążą się z podejściem obiektowym do opisu rzeczywistości. Techniki obiektowe pojawiły się niedawno, najpierw w technologiach programowania np. SMALLTALK, a następnie w organizowaniu baz danych, np. technologia O2. W ogóle sam pomysł obiektowego ujmowania (opisywania) rzeczywistości pojawił się przed dwudziestu laty w Norwegii. Tam właśnie, przy konstrukcji języka programowania SIMULA (ang. *simulation language*), zdefiniowano zasadnicze pojęcia techniki obiektowej. Chodzi o taką konstrukcję myślową, która umożliwia modelowanie różnych aspektów rzeczywistości przy pomocy tych samych (dobrze poznanych i sprawdzonych) elementów. W klasycznym podejściu programy były pisane tak, aby rozwiązać konkretne problemy. Stąd, aby ulepszyć istniejące rozwiązanie, łatwiej było napisać nowe procedury, niż przekształcać istniejące. Programowanie obiektowe omija te (i inne) niedogodności. Oferuje, np. poprzez dziedziczenie, mechanizmy umożliwiające składanie programów na różne sposoby z klas już istniejących. Bazy obiektowe bardziej wyraziście oddają teorię zdarzeń. Wyróżnia się trzy podstawowe warstwy: przechowywanie obiektów, przechowywanie złożonych informacji oraz budowa inteligentnych baz danych. Chodzi o przechowywanie i odczytywanie w pełni ukształtowanych obiektów (patrz: zdarzenie — transakcja gospodarcza) bez redukcji ich do wymuszonego formatu (patrz: rekord bazy relacyjnej). Ważna jest tu możliwość modelowania struktur wynikowych z cząstek elementarnych, gdzie np. zmiana wartości pierwotnych widziana jest we wszystkich rozwinięciach systemu.

Zadaniem tego rozdziału - być może zbyt kontrowersyjnym dla osób ściśle związanych z rachunkowością — jest przedstawienie propozycji rozwiązań z zakresu organizacji struktur informacji wejścia, organizacji baz danych w komputerze i strategii przetwarzania zintegrowanego, informatycznego systemu rachunkowości. Celem tych rozważań jest usunięcie sprzeczności między potrzebami informacyjnymi zarządzania a metodologią rachunkowości w klasycznym jej ujęciu. Chodzi o integrację informacji wejścia w postaci standaryzowanych dokumentów zdarzeń dla księgowości finansowej i materiałowej, utworzenie instytucji *zlecenia* dla równoczesnego opisu stanu kartotek i technologii wytwarzania, zaprojektowanie stosownych baz danych oraz przedstawienie sposobu korzystania z sieci komputerowych w architekturze klient/serwer.

8.1. Integracja dziedzinowa

Rachunkowość stanowi oparty na dokumentach, zamknięty i bilansujący się system ewidencji gospodarczej. Jest ona częścią składową systemu ewidencyjnego, która ujmuje w sposób systematyczny i ciągły przebieg oraz skutki procesów gospodarczych występujących w jednostce gospodarującej, tj. procesów zaopatrzenia, produkcji i zbytu (J. Matuszewicz, 1987). Przedmiotem zainteresowania rachunkowości są zasoby majątku rzeczowego i finansowego, koszty i przychody oraz wyniki działalności⁷². W rachunkowości wykorzystuje się określone metody, reguły postępowania, urządzenia ewidencyjne i środki techniczne umożliwiające rejestrację stanu i zmian majątku jednostki, źródeł jego finansowania (pochodzenia), dokonywanych operacji gospodarczych i wyników działalności⁷³.

Rachunkowość jest także często postrzegana jako proces transformacji informacji, który obejmuje ciąg następujących po sobie czynności, począwszy od obserwacji, poprzez gromadzenie danych, dokonywanie zapasów i ich analizowanie. W końcowym etapie następuje przekazywanie informacji ich użytkownikom. Dostosowywanie rachunkowości do informacyjnych potrzeb zarządzania powoduje konieczność gromadzenia ogromnej ilości danych, szybkiego do nich dostępu, stosowania skomplikowanych algorytmów przetwarzania informacji i różnorodnych form ich prezentacji. Wymagania te przesądzają o celowości zastosowania techniki komputerowej (E. Nowak, 1994).

O zakresie zastosowań informatyki w rachunkowości w polskich przedsiębiorstwach decyduje z jednej strony poziom nasycenia środkami technicznymi i organizacyjnymi, z drugiej natomiast potrzeby systemów zarządzania.

Spośród coraz większej liczby informatycznych systemów zarządzania oferowanych przedsiębiorstwom przez firmy zajmujące się oprogramowaniem (ang. *softwar*), największym zainteresowaniem cieszą się obecnie systemy ewidencyjne, głównie finansowo-księgowo, a także systemy służące rachunkowości: gospodarki magazynowej, zatrudnienia i płac, gospodarki środkami trwałymi oraz systemy związane z obowiązującym systemem podatkowym (E. Nowak, 1994).

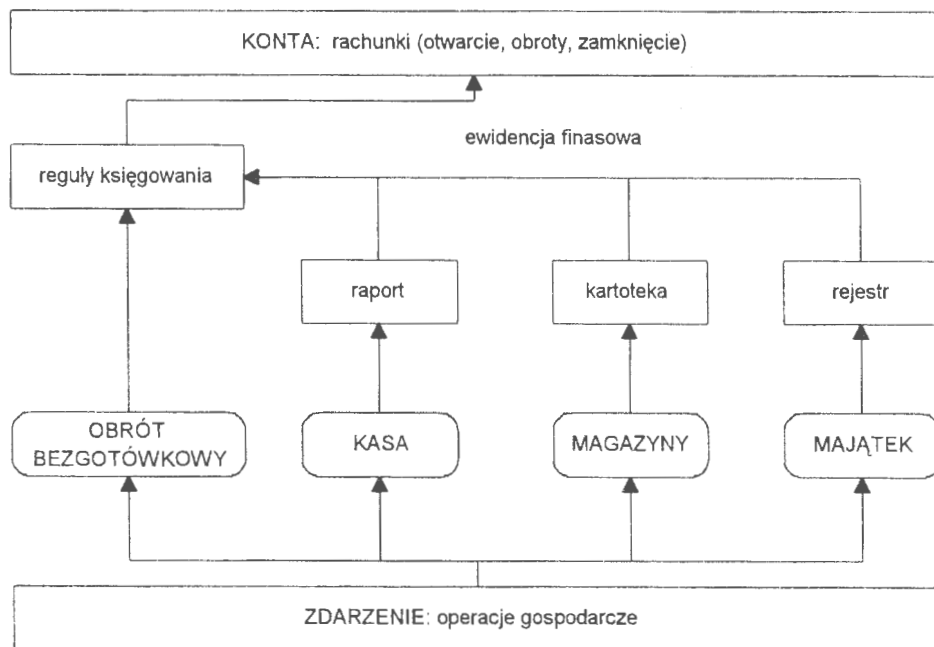
8.1.1. Model systemu informacyjnego

Komputeryzacji rachunkowości towarzyszy integrowanie we wspólnych bazach danych ewidencji księgowej z ewidencją techniczną, handlową czy kadrową. Tego typu zastosowania informatyki określa się jako systemy informatyczne rachunkowości (SIR). Problem brzmi: jak połączyć nieskończoną w zasadzie liczbę dziedzin w jeden wzajemnie powiązany system ewidencji zdarzeniowej. Jednostką czasu byłby chronom {rok, miesiąc, dzień, godzina, minuta, sekunda}. W tym przypadku będziemy dążyć do połączenia w standaryzowanych zbiorach transakcji problemów prowadzenia finansów, kasy i gospodarki materiałowej.

System informatyczny rachunkowości składa się z następujących elementów:

- struktury fizycznej (sprzęt i jego konfiguracja, oprogramowanie podstawowe, oprogramowanie użytkowe),
- wbudowanego pewnego modelu rzeczywistości i odpowiadającej mu struktury informacyjnej oraz procedur obliczeniowych, które odzwierciedlają stany, zdarzenia i procesy gospodarcze,
- otoczenia tego systemu, tj. środowiska materialnego (budynki, instalacje), struktur organizacyjnych (w których system funkcjonuje) oraz związanych z tymi strukturami ludzi (szczególnie dawcy i odbiorcy informacji), systemu kontroli zewnętrznej, systemu ochrony itp.

Zdarzenia gospodarcze dotyczą nie tylko obrotu pieniężnego, ale również zmian ilościowych w magazynach i stanu majątkowego przedsiębiorstwa. W pewnym uproszczeniu można tu mówić również o ewidencji pracy. Mamy więc do czynienia z dziedzinową klasyfikacją, z nieskończonym w zasadzie opisem dziedzin (może ich być bardzo wiele) w systemach informatycznych rachunkowości (księgowości).



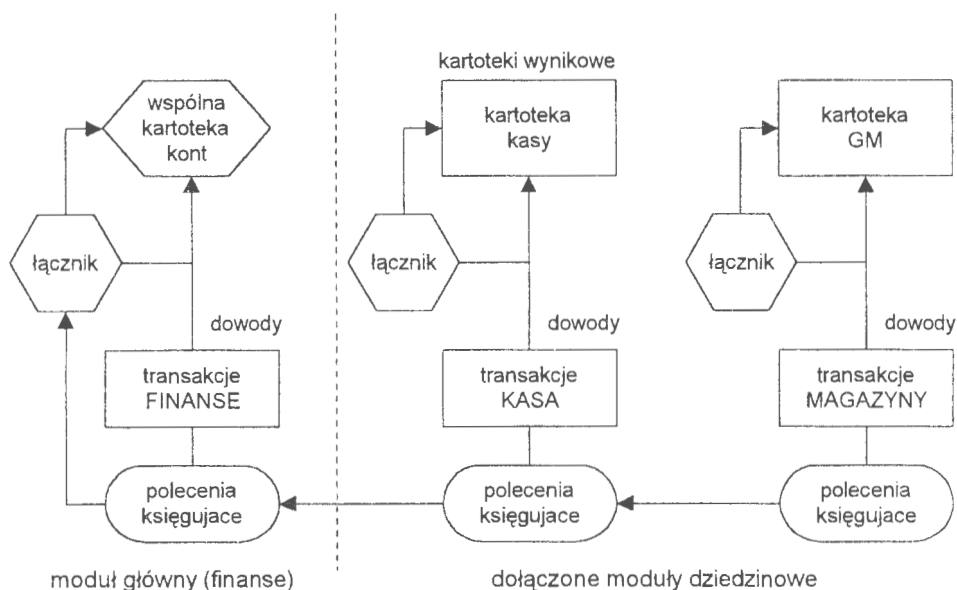
Schemat 22. Uproszczony schemat ewidencji zdarzeń gospodarczych
(Źródło: opracowanie własne)

8.1.2. Jednostka ewidencji zdarzeń

Zdarzenie jest zapisem momentu przeprowadzenia operacji gospodarczej. Omówiono to na przykładzie ewidencji finansowej. Rozwinięcie tego przykładu poza ewidencję finansową pokazuje, że prawie każda operacja gospodarcza mająca swoje źródła poza finansami „kończy swój żywot” właśnie w finansach (zapisie na kontach księgowych).

Na przykład wydano towar z magazynu. Zostaje wystawiony dokument wydania, następuje ilościowa aktualizacja kartoteki magazynowej. Można tę operację wzbogacić również rozliczeniem wartościowym. Następnie trzeba uznać i obciążyć określone konta w systemie finansowym przedsiębiorstwa. Praktycznie rozliczenia finansowe odbywają się w różnych terminach. Najczęściej są to rozliczenia okresowe. Problem automatyzacji prowadzenia ewidencji gospodarek materiałowych jest powszechnie znany i nie będzie dalej przedmiotem rozważań. Interesuje nas sam moment powiązań księgowości materiałowej z księgowością finansową w oparciu o jednolity dokument (komputerowy) ewidencji zdarzenia. Samo połączenie, na które konta należy przenieść

wartości transakcji, można zorganizować bezpośrednio przy pomocy reguł automatycznego księgowania. Nie oznacza to wykluczenia możliwości księgowania autorskiego pojedynczych dowodów (lub ich grup — okresowo) w księgowości finansowej.



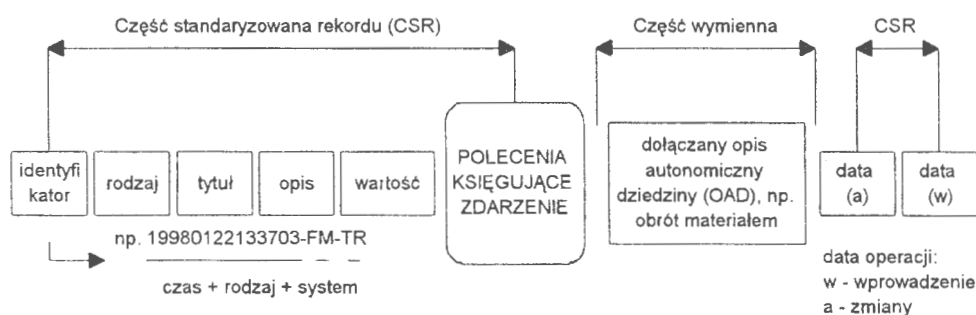
Schemat 23. Sieć temporalnych baz danych systemu rachunkowości

(Źródło: opracowanie własne)

Lista jednostek podstawowych (transakcji — encji) poszerzy się zatem o zapisy związane z obrotem towarowym z równoczesnym wskazaniem, na które konta należy przenieść wartości transakcji w księgowości finansowej. Opisy konkretnych operacji księgowania transakcji materiałowej proponuje się zawrzeć na polu MEMO, co dokładnie przedstawiono w poprzednich rozdziałach. Podobnie rzecz będzie dotyczyła z wpływów i wydatków z kasy oraz rejestracji zmian w stanie majątkowym i przenoszenia wartości amortyzacji na miejsca powstawania kosztów. Problem sprowadza się do wyspecyfikowania typowej dla danego problemu części „dołączanej” rekordu w transakcji. Układ pól całego rekordu składa się zatem z dwu części: standardowej i typowej dla danego zdarzenia, np. specyfikacja walut w obsłudze kasy lub opis obrotu towarowo-materiałowego w magazynie.

Wprowadzone np. parametry gospodarki materiałowej mają za zadanie otworzyć kartotekę materiałów w magazynie i aktualizować ją w zakresie: przychodów i rozchodów wewnętrznych (np. na produkcję) oraz przychodów

i rozchodów na zewnątrz (np. na sprzedaż). Punktem odniesienia jest oznakowany magazyn, z którego przeprowadza się rozchody i w którym ewidencjonuje się przychody materiałowe. Lista dodatkowych informacji z tym związanych to: RD — *rodzaj operacji magazynowej* (oznakowanie w identyfikatorze), KTM — *kod towarowo-materiałowy* (może być także SWW lub inne), NAZWA — *nazwa potoczna materiału*, ILOŚĆ „I” — *ilość pierwsza* (główna), JM „I” — *jednostka miary* (pierwsza), ILOŚĆ „II” — *ilość druga* (dodatkowa), JM „II” — *jednostka miary* (dodatkowa), CENA — *cena jednostkowa* (zł/jedn. pierwsza), KOD — *kod ceny jednostkowej* (pierwszej) i MG — *symbol magazynu*, w którym dokonano obrotu.



Rys. 11. Standaryzacja opisu zdarzenia gospodarczego w zintegrowanym systemie (Źródło: opracowanie własne)

Wprowadzenie przedstawionych parametrów gospodarki materiałowej umożliwi bieżące prowadzenie kartoteki magazynowej. Przy identyfikacji towarów możliwe są dwa warianty: każdy magazyn posiada odrębną kartotekę (o symbolu MG) lub identyfikator magazynu będzie dołączany do każdego towaru. Powstały identyfikator: RR — *rok kalendarzowy*, MG — *magazyn w przedsiębiorstwie* i KTM — *towar w magazynie*, wydaje się rozwiązaniem lepszym do kontroli wszystkich towarów w magazynach równocześnie. Rok kalendarzowy podaje się, podobnie jak w przypadku kont, dla rozróżnienia stanów modelowanych i stanu aktualnego. Po zaktualizowaniu stanu materiałowego należy uzupełnić transakcje o klasyfikatory zapisu na kontach w księgowości finansowej. Jak już wspomniano, można to uczynić przy pomocy reguł automatycznego księgowania, a w szczególnych przypadkach przy pomocy księgowania autorskich. Mogą wystąpić zdarzenia pozorne, np. określenia planu do realizacji. Tego rodzaju normatywy mogą być zawarte w kartotekach

wynikowych, a źródła tych stanów powinny być umieszczone w zbiorze dowodów (transakcji). W ten sposób można dołączać w zasadzie nieskończoną ilość dziedzin do jednego (zintegrowanego) systemu ewidencji gospodarczej.

8.1.3. Instytucja zlecenia

Oznakowanie instytucji zlecenia, które celowo wprowadzono do identyfikatora zdarzenia gospodarczego, ma znaczenie zasadnicze i nowatorskie. We wstępie postawiono tezę o *nienadążaniu rozwiązań informacyjnych, a także informatycznych rachunkowości za prowadzoną działalnością gospodarczą przedsiębiorstwa*. Serwisy informacyjne dotyczą przeważnie zaszłości w cyklach comiesięcznych, rocznych. Brak jest w nich możliwości pełnego odwoływania się do stanu rzeczywistości takiej, jaką ona była w przeszłości i może być, z pewnym prawdopodobieństwem, w przyszłości. Integracja księgowości finansowej, materiałowej i ewidencji majątku umożliwia pełniejsze postrzeganie zdarzenia gospodarczego.

Wprowadzenie do systemu jednorodnego zapisu tegoż zdarzenia pozwala na:

- wyjaśnienie sprzeczności między potrzebami informacyjnymi prowadzenia biznesu a formalnymi zasileniami informacyjnymi komputerowych systemów rachunkowości,
- umożliwienie ewidencji technologii produkcji w przedsiębiorstwie, co jest związane z rozwojem metod MRP i reengineeringu,
- prowadzenie pełnego rozrachunku dla wybranego kierunku produkcji lub wprowadzonego zadania do wykonania.

Istotną zaletą jest możliwość wielokierunkowego przedstawiania przebiegu (historii), np. rozpoczętej inwestycji, nie tylko w wartościach finansowych (co można było osiągnąć w opcji POTWIERDZENIE), ale również w wymiarach rzeczowych. Pod symbolem określonego zadania inwestycyjnego (zlecenia) możemy, dla dowolnego przedziału czasu lub innego kryterium, zestawiać dane na różne sposoby edycyjne. Rzecz jasna prezentowany język użytkownika musi ulec wzbogaceniu o opcje opisujące stany materiałów i ponoszonych kosztów majątkowych.

Ważnym zastosowaniem przedstawionego podejścia — *instytucji zlecenia* — jest rachunkowość rolnicza, chyba najtrudniejsza do poprawnego ewidencjonowania — produkcja. Gospodarstwa rodzinne lub małe przedsiębior-

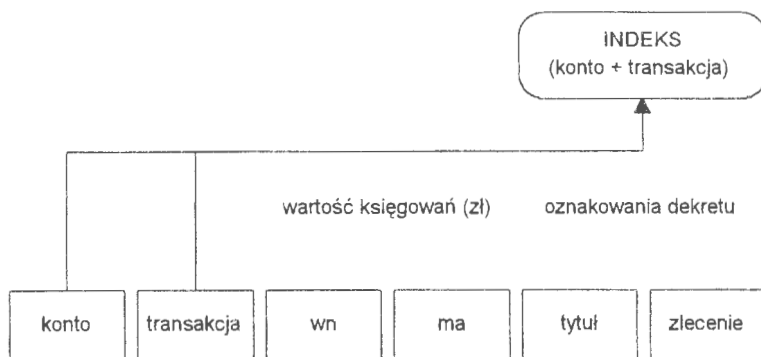
stwa dzierżawców rolnych nie stać na prowadzenie pełnej rachunkowości. Rzeczywistość może być taka, że na niewielu kontach tekstowych (np. przy pomocy reguł automatycznego księgowania) prowadzi się bieżący rachunek kosztów i przychodów, a dla głównego kierunku produkcji, np. dla produkcji mleka (obora), ustanawia się zlecenie i rejestruje wszystkie operacje gospodarcze (np. nakłady pasz na produkcję mleka i przychody z tym związane). Rolnik następujące korzyści: zna wynik finansowy gospodarstwa (jako całości) i bardzo szczegółowo zna ekonomikę i technologię głównego kierunku, tj. produkcji mleka. Najważniejsze, że w sposób nieskomplikowany może prowadzić pełną rachunkowość w domu lub usługowo w specjalistycznych biurach rachunkowości rolnej.

8.1.4. Reprezentacja czasu

W przypadku samej księgowości finansowej w przeciętnym przedsiębiorstwie, transakcji jest ok. 30 tys. i około 3 tys. kont. W związku z tym ŁĄCZNIK (relacje konto ↔ transakcje) musi mieć około 100 tys. rekordów. Wprowadzenie np. obrotu towarowego kilkakrotnie zwiększy ilość transakcji i rekordów łącznika. Dla sprawnego działania zwykłe zbiory klasy DBF mogą tu nie wystarczyć. Trzeba odwołać się do wyspecjalizowanych serwerów danych i metod organizujących dostęp ze względu na kryterium czasu. Interesującym rozwiązaniem mogłoby być programowanie systemu, np. w plikach SQL i przetwarzanie danych w oparciu o technologię klient/serwer. Przez protokół ODBC do przetwarzania można wykorzystać serwery dużej mocy, np. ORACLE, INFORMIX lub SYBASE. Nieistotne stałoby się ciągłe pamiętanie o ochronie danych, sprawdzanie plików danych z tabelami indeksów i problemy zakleszczeń w przetwarzaniu sieciowym. Możliwych jest tu kilka rozwiązań: ustanowieniem łączników cząstkowych (dla każdej z dziedzin oddzielnie lub posłużenie się jednym (wspólnym) zbiorem ŁĄCZNIKA, co przedstawiono na schemacie 22). Istotny jest przy tym fakt, że dla danych księgowości materiałowej (i innych dziedzin poza finansowych) nie istnieje konieczność tworzenia specjalnego łącznika. W przypadku np. gospodarki materiałowej funkcje te pełni tablica indeksów o łączonym kluczu IDENTYFIKATOR + KTM. Są to bezpośrednio pola rekordu transakcji, z których automatycznie można tworzyć (i odtwarzać) tablice o różnych kombinacjach kluczy indeksujących w momencie wprowadzania danych do systemu.

Problemem pozostaje szybkość wyznaczania stanów dla wskazanych przedziałów czasu. W księgowości finansowej ustanowiono ŁĄCZNIK, przy po-

mocy którego selekcjonowano transakcje ze względu na przedział czasu i konta. Eliminowało to konieczność przeglądania całego pliku transakcji. Korzyść z oszczędności czasu była tu niewątpliwa. Przykładowo, dla zadania z przedziałem czasu obejmującym wszystkie konta trzeba przejrzeć zbiór wszystkich transakcji tyle razy, ile jest kont w zadaniu rachunkowości. Następnie z wybranych transakcji należy obliczyć bilans otwarcia na wskazany przedział czasu, obroty i bilans zamknięcia. ŁĄCZNIK eliminuje tu przeglądanie, ale nie zwalnia od obliczeń bilansu otwarcia, obrotów i bilansu zamknięcia przeprowadzanych bezpośrednio na transakcjach.



Schemat 24. Organizacja rozszerzonego zbioru ŁĄCZNIKA w zadaniu
(Źródło: opracowanie własne)

Mozna problem ten rozwiązać następująco. Do struktury ŁĄCZNIKA (konto, transakcje) trzeba dołączyć dodatkowe pola: wartości operacji gospodarczej dla strony Wn lub Ma oraz oznaczenie tytułu i zlecenia; tworzymy stany dla chronomu (punktochwili): wartość ostatniej operacji i jego cechy. Jeżeli zostanie zadane pytanie „podaj stany finansów w przedziale czasu, np. 5 — 7. 01.”, to dostępy bezpośrednie, wykonane przy pomocy polecenia SEEK, udostępnią dwa stany tego samego konta: bilans otwarcia na dzień 5.01. (Wn i Ma) i bilans zamknięcia na dzień 7.01. (Wn i Ma). Osiąga się to poprzez zsumowanie wartości Wn i Ma bezpośrednio na pliku ŁĄCZNIKA, co jest zabiegiem znacznie szybszym niż praca na transakcjach. Różnice tych stanów wyznaczają obroty na wskazanym koncie. Natomiast specyfikacja stanów operacji Wn i Ma dla poszczególnych zapisów wyznacza historię konta dla wskazanego przedziału czasu.

Integracja w rachunkowości została przedstawiona jako możliwość wprowadzenia zasadniczych danych źródłowych w postaci standaryzowanego ciągu rekordów transakcji. Rozwinięcie tego ciągu, podobnie jak w księgowości finansowej, powinno utworzyć i aktualizować kartoteki kont, kas, magazynów i na bieżąco prowadzić rejestr umorzeń środków trwałych. Problem drugi to skrócenie okresu obrachunkowego do jednego dnia (a nawet jednej sekundy). Może to z punktu widzenia comiesięcznej sprawozdawczości państwowej i skarbowej jest nieistotne. Przedstawiony projekt organizacji bazy dokumentów źródłowych — zdarzeń w postaci standaryzowanych rekordów — umożliwia jednak tego rodzaju operacje i w jego oparciu można rozbudowywać (do bardzo dużych rozmiarów) system informatyczny. Wiele funkcji standaryzujących przeksięgowania można by powierzyć regułom automatycznego księgowania. Reguły mogłyby całościowo wykonywać operacje związane z zamykaniem okresów obrachunkowych. Tego rodzaju rozwiązania na pewno pojawią się w niedalekiej przyszłości. Użycie do tego celu rozwiązań relacyjnych ma jednak pewne mankamenty, głównie z punktu widzenia szybkości dostępu do wyników dla różnych momentów czasu i różnych warunków. Chodzi między innymi o wersje połączenia dokumentów źródłowych w naturalną hierarchię podporządkowania (również z atrybutami przyzwolenia), jak ma to miejsce w rzeczywistości.

8.2. Model obiektowy rachunkowości

Rozpatrzmy niektóre problemy opisu rzeczywistego systemu rachunkowości w powszechnie przyjętych standardach obiektowych. Podejście obiektowe zrodziło się w latach 60. i właściwie dopiero w końcu tego stulecia nabrało znaczenia w informatyce. Bezspornie zdobyło sobie uznanie w programowaniu, wypierając klasyczne programowanie proceduralne. Natomiast w bazach danych wprowadzanie w miejsce relacyjnych rozwiązań technologii obiektowych napotyka na szereg trudności. Wydaje się, że na przeszkodzie stoi dość trudna interpretacja w definiowaniu obiektowych baz danych wobec dużego dorobku praktycznych rozwiązań baz relacyjnych. Bazy relacyjne są dwuwymiarowymi tablicami danych, składającymi się z wierszy (rekordy) i kolumn (pola). Powiązania między tablicami tworzą relacje, umożliwiające przenoszenie (dziedziczenie) danych z różnych plików. W bazach obiektowych powiązania te są daleko bardziej złożone. Szacuje się, że bazy obiektowe stanowią obecnie już kilka procent w oprogramowaniu i eksploatacji baz danych.

8.2.1. Podstawowe pojęcia

Obiektem może być każda encja (cząstka) pochodząca ze świata rzeczywistego i posiadająca jednoznaczny wartość identyfikacyjną w całym systemie. Obiekt charakteryzuje się jednym lub więcej atrybutami, jedną lub więcej metodami, które operują na wartościach tych atrybutów. Atrybuty obiektów są analogiami atrybutów (wartości kolumn) w relacyjnych bazach danych. Metody w obiektach są analogiami procedur dostarczanych w niektórych systemach hierarchicznych baz danych lub w językach programowania. Zasadniczym pojęciem w obiektowych bazach danych jest hierarchia i dziedziczenie. Wszystkie obiekty, które mają ten sam zbiór atrybutów mogą być zgrupowane w jednej klasie. Klasy w systemie baz danych tworzą hierarchię zwaną hierarchią klas. W niektórych systemach obiektowych zakłada się, że każda klasa może mieć tylko jedną klasę, która jest jej bezpośrednim poprzednikiem i nazywa się wtedy nadklasą. Inne rozwiązania definiują, że każda klasa może mieć wiele nadklas. W pierwszym przypadku klasa dziedziczy atrybuty i metody tylko od jednej nadklasy, w drugim przypadku mamy do czynienia z wielokrotnym dziedziczeniem metod i atrybutów. Atrybuty i metody określone w tej klasie są następnie dziedziczone w sposób rekurencyjny przez wszystkie jej podklasy.

Podstawowe pojęcia, przyjęte w obiektowej bazie danych za Von Kimem⁷⁴, możemy wyspecyfikować w następujący sposób:

- *obiekt i identyfikator* obiektu jest jednostką podstawową bazy danych; każdy obiekt może być rekurencyjnie powiązany z innymi obiektami przez związki semantyczne; miejscem odwołań do innych obiektów są atrybuty; mówi się, że obiekt składa się z identyfikatorów obiektów, a identyfikatory obiektów są jedynymi miejscami dostępu do atrybutów im przypisanych,
- *atrybuty i metody* opisują stan obiektu i sposoby jego zmiany; atrybut może mieć wartość pojedynczą lub reprezentować zbiór wartości ze swojej dziedziny, zwanych dalej klasą; natomiast metody są procedurami, przy pomocy których wywołujemy zmiany i informujemy się o stanie obiektu,
- *klasę* stanowią wszystkie obiekty, które mają ten sam zbiór atrybutów i metod; każdy obiekt należy tylko do jednej klasy i jest jej instancją (lub konkretem); pojęcie klasy jest zasadniczym elementem łączącym obiektowe systemy i bazy danych,
- *hierarchia klas i dziedziczenie* tworzy aktywny i ściśle ukierunkowany układ przepływu danych i metod interpretacji w obiektowej bazie danych.

Przykładowo, każda klasa o wyższym poziomie dziedziczy wszystkie atrybuty oraz metody podklas i odwrotnie: wszystkie atrybuty i metody określone dla nadklasy są dziedziczone w sposób rekurencyjny przez wszystkie podklasy.

Obiektowa baza danych może być reprezentowana przy pomocy tzw. zbioru „systemowo” zdefiniowanych klas, zbioru instancji obiektowych oraz zbiorów indeksowych (tablic kluczy), umożliwiających szybki dostęp do bazy danych. W zbiorze systemowo zdefiniowanych klas (SZK) przechowywane będą:

- klasy występujące w obiektowej bazie danych,
- atrybuty zdefiniowane dla każdej klasy,
- metody zdefiniowane dla każdej klasy,
- związki agregacji zachodzące między klasami i atrybutami,
- związki generalizacji zachodzące między klasą a jej podklasami.

Jeżeli klasy tworzą hierarchię wielokrotnego dziedziczenia, to tworzą one zakorzeniony, spójny graf acykliczny, często zwany kratą klas, w którym istnieje jeden korzeń (klasa specjalna) — R. Gupta, E. Horowitz (1991), K.B. Douglas (1996).

8.2.2. Model konceptualny bazy

Obiektowa baza danych (OBD) według B. Śmiałkowskiej, jest uporządkowaną szóstką elementów⁷⁵. Oznaczmy to symbolicznie w sposób następujący:

$$OBD = \langle K, M, O, A, DH, KK \rangle \quad [8.01]$$

W definicji tej

$$K = \{K_1, K_2, \dots, K_k\} \quad [8.02]$$

jest zbiorem klas (K_i — oznacza i -tą klasę w zbiorze klas),

$$M = \{M_1, M_2, \dots, M_m\} \quad [8.03]$$

jest zbiorem metod,

$$O = \{O_1, O_2, \dots, O_p\} \quad [8.04]$$

to zbiór obiektów, w którym zdefiniowano atrybuty ze zbioru atrybutów

$$A = \{A_1, A_2, \dots, A_n\}, \quad [8.05]$$

DH — jest drzewem hierarchii klas, którego wierzchołkami są klasy, a gałęzie tego drzewa zdefiniowano relacją bycia „podklasa” (lub „nadklasa”) klasy. Element KK w definicji OBD określa tzw. kompozycję klas, która zostanie zdefiniowana poniżej.

Każda metoda $M_i \in M$ jest przekształceniem zbioru parametrów wejściowych:

$$PI_i^1, PI_i^2, \dots, PI_i^s \quad [8.06]$$

w zbiór parametrów wyjściowych:

$$PO_i^1, PO_i^2, \dots, PO_i^r \quad [8.07]$$

co symbolicznie zapiszemy w następujący sposób:

$$\begin{aligned} M: \text{dom}(PI_i^1) \times \text{dom}(PI_i^2) \times \dots \times \text{dom}(PI_i^s) \rightarrow \\ \rightarrow \text{dom}(PO_i^1) \times \text{dom}(PO_i^2) \times \dots \times \text{dom}(PO_i^r) \end{aligned} \quad [8.08]$$

gdzie $\text{dom}()$ oznacza dziedzinę, a symbol \times iloczyn kartezjański.

Dla każdej klas $K_i \in K$ w OBD definiuje się kompozycję klasy KK jako wartość jednoznacznej funkcji zdefiniowanej następująco:

$$KK: K \rightarrow AxM \quad [8.09]$$

gdzie

$$KK(K_1) = (A_1^1, A_1^2, \dots, A_1^m, M_1^1, M_1^2, \dots, M_1^m) \quad [8.10]$$

$$A_i^l \in A \quad i \quad M_i^q \in M \quad [8.11]$$

dla $l = 1, 2, \dots, m$ i $q = 1, 2, \dots, m$ oraz $i = 1, 2, \dots, k$.

Drzewo hierarchii klas DH jest dwójką uporządkowaną:

$$DH = \langle K, V \rangle \quad [8.12]$$

gdzie K jest zbiorem klas w OBD a V jest zbiorem gałęzi drzewa. Gałęzie drzewa hierarchii definiuje się relacją „bycia” nadklasą klasy.

Powiemy, że klasa $K_i \subset K$ jest nadklasą klasy $K_w \in K$, co zapiszemy symbolicznie:

$$K_w \subset K_i \quad [8.13]$$

wtedy i tylko wtedy, jeżeli:

$$KK(K_i) = (A_i^1, A_i^2, \dots, A_i^{in}, M_i^1, M_i^2, \dots, M_i^{im}) \quad [8.14]$$

$$KK(K_w) = (A_w^1, A_w^2, \dots, A_w^{wn}, M_w^1, M_w^2, \dots, M_w^{wm}) \quad [8.15]$$

to:

$$\begin{aligned} \exists_{q \in \{1, 2, \dots, wm\}} \exists_{i \in \{1, 2, \dots, in\}} \text{dom}(A_w^q) = \text{dom}(A_i^i) \wedge \\ \wedge \{M_w^1, M_w^2, \dots, M_w^{wm}\} \supseteq \{M_i^1, M_i^2, \dots, M_i^{im}\} \end{aligned} \quad [8.16]$$

Jeżeli $K_w \subset K_i$, to K_w w nazywa się podklasą klasy K_i .

Dla pełnej definicji drzewa hierarchii bazy obiektowej należy określić zbiór gałęzi drzewa V , definiując go następująco:

$$V = \left\{ (K_j, K_w) : K_w \subset K_i \wedge K_i \in K \wedge K_w \in K \right\} \quad [8.17]$$

Powiemy, że obiekt $O_j \in O$ jest instancją klasy $K_i \in K$, co symbolicznie zapiszemy:

$$O_j = \text{inst}(K_i) \quad [8.18]$$

wtedy i tylko wtedy, jeżeli:

$$O_j = (a_j^1, a_j^2, \dots, a_j^r, m_j^1, m_j^2, \dots, m_j^s) \quad [8.19]$$

$$KK(K_j) = (A_j^1, A_j^2, \dots, A_j^r, M_j^1, M_j^2, \dots, M_j^s) \quad [8.20]$$

to:

$$a_j^1 \in \text{dom}(A_j^1) \wedge \forall_{i=1, 2, \dots, r} \forall_{q=1, 2, \dots, s} m_j^q \in \{M_i^1, M_i^2, \dots, M_i^s\} \quad [8.21]$$

Jeżeli A^q jest atrybutem obiektu O_j , a M^s jest metodą tego obiektu, to:

$$A^q = atr(O_j) \quad [8.22]$$

$$M^s = meth(O_i) \quad [8.23]$$

Należy zauważyć, że jeśli:

$$O_i = inst(K_j) \quad [8.24]$$

$$A^q = atr(O_i) \quad [8.25]$$

$$M^s = meth(O_j) \quad [8.26]$$

to:

$$A^q \in KK(K_j) \quad \text{oraz} \quad M^s \in KK(K_j) \quad [8.27]$$

Klasa K_i jest komponentem klasy , co symbolicznie zapiszemy:

$$K_i = comp(K_j) \quad [8.28]$$

wtedy i tylko wtedy, gdy:

$$\exists_i O_i = inst(K_j) \quad [8.29]$$

oraz:

$$\exists_p A_p = atr(O_j) \wedge A_p = adr(K_p) \wedge \exists_s M^s = meth(O_i) \wedge M^{s'} = meth(K_j) \quad [8.29]$$

W przypadku wielwersyjnych baz danych dodatkowo należy wprowadzić drzewo wyvodu wersji DD . Jest ono dwójką uporządkowaną w postaci:

$$DD = \langle ZK, ZV \rangle \quad [8.31]$$

gdzie ZK jest zbiorem obiektów wersjowanych lub zbiorem wersjowanych baz danych, a ZV oznacza gałęzie drzewa wyvodu wersji. Drzewo to jest definiowane w zależności od przyjętego modelu wersjowania.

8.2.3. Organizacja klas danych

Bazy obiektowe dzięki swemu podobieństwu do baz relacyjnych mogą być przez nie zastępowane i odwrotnie. Przewaga baz obiektowych tkwi w potencjale intelektualnym, który można wykorzystać przy opisie i interpretacji rzeczywistości. Przedstawiono jeden z możliwych do przyjęcia wariantów orga-

nizacji obiektowej bazy danych dla rachunkowości (rys.12). Przy projektowaniu tego schematu korzystano z gotowych rozwiązań ewidencji zdarzeń gospodarczych z poprzednich rozdziałów, której praktyczna realizacja w konwencji relacyjnych baz danych miała miejsce już w 1992 roku, jako pierwszej w Polsce temporalnej bazy finansów firmy (patrz: R. Budziński, W. Gos, B. Nadolna, 1992). Na rysunku tym podwójną kreską oznaczono hierarchię dziedziczenia klas, a pojedynczą hierarchię kompozycji klas. W prostokątach ujęto klasy z określonymi dla tych klas atrybutami. Prostokąty puste dla odpowiednich klas dziedziczą atrybuty z nadklas hierarchii dziedziczenia. Każdy węzeł w takim grafie jest dostępny z korzenia. W systemie z pojedynczym dziedziczeniem mamy hierarchię klas, którą można opisać drzewem posiadającym jeden korzeń (być może specyficznie definiowanym). Hierarchia klas reprezentuje związek tzw. generalizacji między klasą a jej pośrednimi i bezpośrednimi podklasami. Każda klasa dziedziczy wszystkie atrybuty i metody swoich nadklas; może dodatkowo charakteryzować się własnymi metodami i atrybutami, co dodatkowo rozszerza możliwości opisu rzeczywistości.

Hierarchia klas

Za punkt wyjścia w przedstawionym rozwiązaniu przyjęto zdefiniowanie klasy *Zdarzenie Gospodarcze*, która ma stanowić podstawowy zapis momentu przeprowadzenia operacji gospodarczej. Klasa ta ma trzy atrybuty: *identyfikator*, *źródło* i *księgowanie*. Dziedzinaми tych atrybutów są odpowiednio następujące klasy: *Transakcja*, *DokumentŹródłowy* i *OperacjeKsięgowe*. Z kolei dziedzinami atrybutów tych klas są kolejne zdefiniowane klasy lub (już tylko) klasy wartości pierwotnych (np. integer, string itd.). Jeśli dziedziną atrybutu jest klasa pierwotna, to wartość atrybutu przechowywana w bazie danych jest instancją lub zbiorem instancji dziedziny. Jeśli dziedziną atrybutu nie jest klasa pierwotna, to wartość przechowywana dla atrybutu jest identyfikatorem (odwołaniem, wskaźnikiem) lub zbiorem identyfikatorów instancji dziedziny. Dla lepszego zrozumienia tej problematyki należy omówić kolejno hierarchie klas, których początek (korzeń) ma miejsce w klasie *ZdarzenieGospodarcze*.

Klasa *Transakcja* (dziedzina atrybutu *identyfikator*) posiada dwa atrybuty: *czas* — chronom zdarzenia, *miejsce* — zadanie rachunkowości, których dziedzinami są klasy pierwotne oraz jeden atrybut: *zlecenie*, którego dziedziną jest klasa *Zlecenia*. Jak widać każda instancja klasy *ZdarzenieGospodarcze*, oprócz systemowego identyfikatora OID (ang. *object identifier*), musi być zidentyfikowana przez *czas* — podstawowy element temporalnych baz danych (np. *czas zajścia zdarzenia gospodarczego z dokładnością do jednej sekundy*) oraz przez identyfikator prowadzonego aktualnie zadania rachunkowości. Identyfi-

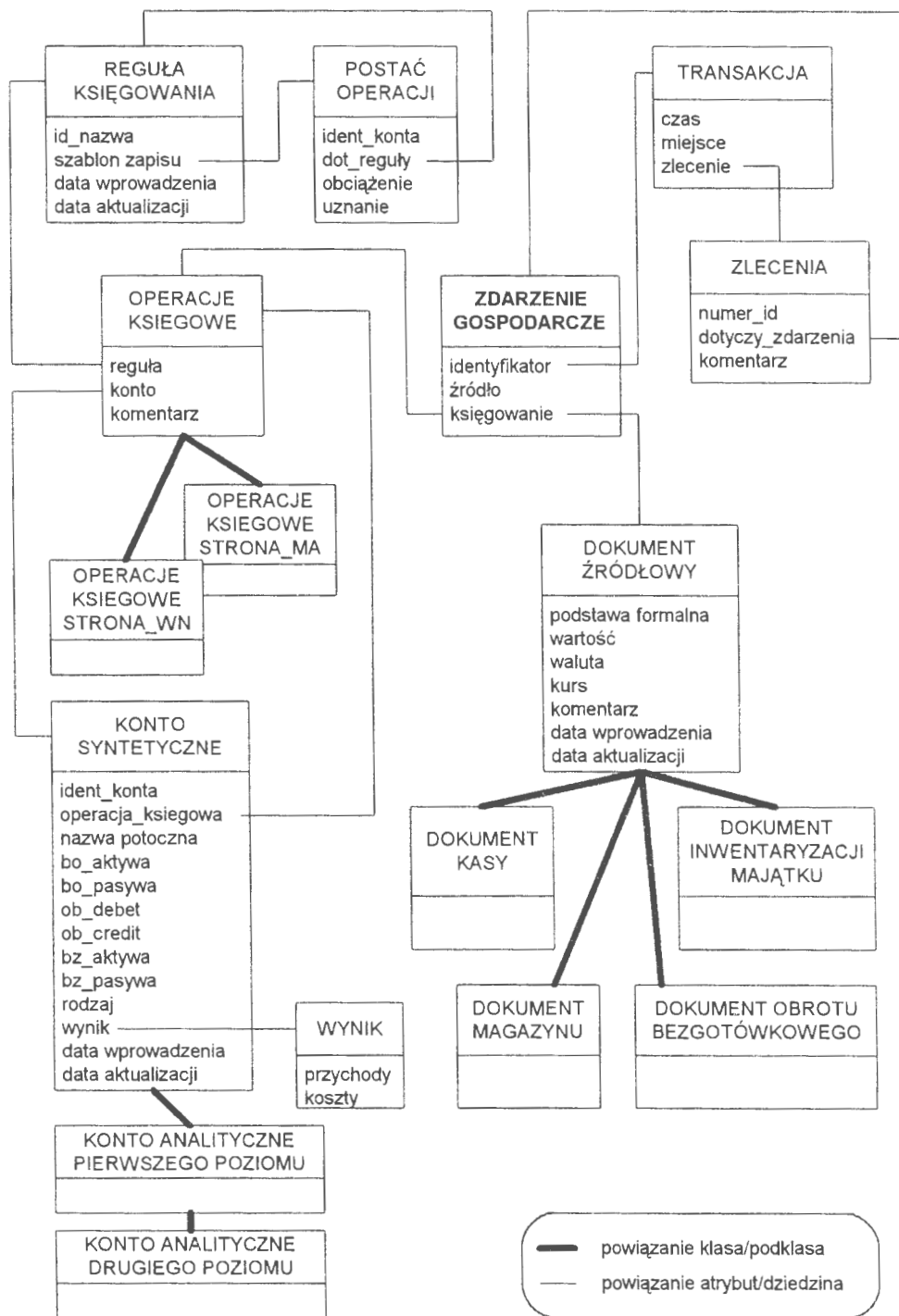
kację tę wzbogacono o opcjonalny wybór zakwalifikowania zdarzeń gospodarczych (transakcji) do grupy objętej zleceniem. Jeśli zostanie założone nowe zlecenie, to musi mu być przyporządkowana chociaż jedna transakcja. Klasa *Zlecenia* ma trzy atrybuty: *numer_id* — numer otwartego zlecenia (klasa pierwotna np. integer), *komentarz* opis zlecenia (klasa pierwotna np. string) oraz *dotyczy_zdarzenia* — wskaźnik systemowy OID instancji klasy *ZdarzenieGospodarcze*. Dziedziną atrybutu *dotyczy_zdarzenia* jest więc klasa *ZdarzenieGospodarcze*. Wynika z tego, że gałąź ta w hierarchii kompozycji klas, w grafie schematu, jest cykliczna. Rozwiązania takie stosuje się do formułowania zapytań wyrażających rekursję, czego nie da się zdefiniować w relacyjnych bazach danych. Przykładem może być zapytanie o „wszystkie zdarzenia gospodarcze z 1997 roku, które dotyczą zleceń”. Intuicyjna składnia takiego zapytania wygląda następująco:

```
select (ZdarzenieGospodarcze :Z)
      (Identyfikator Czas > 1996.12.31:23.59.59 and
      Identyfikator Czas < 1998.01.01:00.00.00 and
      Identyfikator Zlecenie Dotyczy_zdarzenia = Z)
```

W zapytaniu tym zmienna *Z* przyjmuje wartości wszystkich instancji klasy *ZdarzenieGospodarcze* i klasa ta jest docelową klasą zapytania.

Klasa *DokumentŹródłowy* (rys. 12, dziedzina atrybutu źródło) jest korzeniem hierarchii klas z podklasami *DokumentKasy*, *DokumentMagazynu*, *DokumentInwentaryzacjiMajątku*, *DokumentObrotuBezgotówkowego*. Wszystkie te podklasy mogą dziedziczyć atrybuty od swojego przodka lub mogą dodatkowo charakteryzować się własnymi atrybutami. Dopuszcza się tu także dalszą specjalizację (poprzez dziedziczenie) dokumentów źródłowych w danym dziale, czego nie pokazano na rys. 12. Dziedzinaми wszystkich atrybutów klasy *DokumentŹródłowy* są klasy pierwotne. Na dodatkową uwagę zasługuje fakt zastosowania atrybutów domyślnych, których wartość jest używana w każdej instancji klasy, jeżeli użytkownik nie dostarczy bezpośrednio wartości dla tego atrybutu. W przypadku klasy *DokumentŹródłowy* może to być atrybut: waluta = „zł”.

Ostatnim atrybutem klasy *ZdarzenieGospodarcze* jest księgowanie. Dziedzina tego atrybutu jest klasa *OperacjeKsięgowe*, która stanowi korzeń hierarchii klas z podklasami *OperacjaKsięgowaStr_Wn*, *OperacjaKsięgowaStr_Ma*. Podklasy te, jak poprzednio, dziedziczą od swojego przodka wszystkie atrybuty. Dziedzina atrybutu konto jest klasa *KontaSyntetyczne*. Klasa ta może być (jak na rys. 12) korzeniem hierarchii klas, która odpowiada hierarchii w do-



Rys. 12. Model obiektowej organizacji danych w systemie rachunkowości

wolnie przyjętym zakładowym planie kont. Dziedzina kolejnego atrybutu reguła jest klasa *RegułaKsięgowania*. Atrybut ten jest nieobowiązkowy przy wprowadzaniu dowolnej operacji księgowej, gdyż zakłada się, że w programie obsługującym tę bazę danych możliwe będzie księgowanie „ręczne” (tradycyjne) lub z wykorzystaniem wcześniej skonstruowanych „reguł automatycznego księgowania” (przechowywanych jako instancje klasy *RegułaKsięgowania*).

Można np. rozważyć następujące polecenie księgowania: firma ma dwóch pracowników, z kasy wypłacono im pożyczkę. Przyjmuje się, że całkowita kwota wypłacona z kasy wynosi 10 000 zł (dokument źródłowy: wartość = 10 000). Księgowanie mogłoby przebiegać następująco:

1. księgujemy stronę „Ma” (patrz klasa *OperacjaKsięgowaStr_Ma*):

- szukamy atrybutu konto, dla którego `ident_konta = „101.”`, gdzie `nazwa_potoczna = „kasa”` (patrz klasa *KontaSyntetyczne*), po znalezieniu konta uznajemy je całą kwotą: `ob_credit += wartość` (gdzie: `ob_credit` to wartość obrotów konta po stronie „Ma”);

2. księgujemy stronę „Wn” (patrz klasa *OperacjaKsięgowaStr_Wn*):

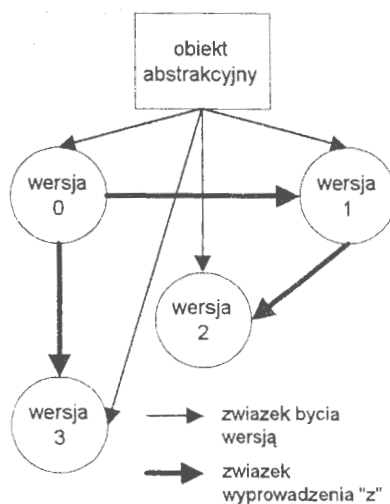
- szukamy atrybutu konto, dla którego `ident_konta = „240.5.01.”`, gdzie `nazwa_potoczna = „Rozrachunki z pracownikiem A”` (patrz klasa *KontaAnalityczneDrugiegoPoziomu*), po znalezieniu konta obciążamy je np. połową kwoty: `ob_debet += 0,5*wartość` (gdzie: `ob_debet` to wartość obrotów konta po stronie „Wn”), następnie dla atrybutu `operacja_księgowa`, którego dziedzina jest klasa *OperacjeKsięgowe* szukamy kolejnego konta, aby dokończyć proces księgowania po stronie „Wn”;
- z obiektu klasy *KontaAnalityczneDrugiegoPoziomu* poprzez rekursję zagłębiamy się w kolejnym koncie, aż do wyczerpania częściowych wartości składających się na całą kwotę stron „Wn”; w naszym wypadku szukamy jeszcze jednego atrybutu konto dla którego `ident_konta = „240.5.02.”`, gdzie `nazwa_potoczna = „Rozrachunki z pracownikiem B”`, po znalezieniu konta obciążamy je pozostałą częścią kwoty: `ob_debet += 0,5*wartość`.

Przedstawiony mechanizm księgowania stosunkowo prosto wskazuje na możliwości księgowania każdych ze stron Wn i Ma niezależnie i wielokrotnie, aż do wyczerpania wartości transakcji i zgodności sum Wn i Ma.

Atrybuty czasu

Podjęcie obiektowe do baz danych stwarza również dodatkowe możliwości obsługi danych czasowych przechowywanych w bazie. Pozwala bowiem stworzyć specjalne metody obsługi i być może przyspieszyć operacje związa-

ne z wykonywaniem zapytań dotyczących danych historycznych obiektów. Jednym ze sposobów rozwiązania tego problemu może być przechowywanie w bazie danych obiektu w formie wielu jego wystąpień. Jest tu pewna analogia do procesu wersyfikacji przedstawionej przez Von Kima⁷⁶. Autor wykazuje, że można wyprowadzić rekurencyjnie dowolną liczbę wersji, uzyskując hierarchię wersji danego obiektu, jeżeli zostanie utworzona najpierw jego wersja pierwotna. W ogólności wystąpienia obiektów z atrybutami czasu mogą być kolejnymi wersjami zdarzenia pierwotnego i wystąpień następnich. Te kolejno pojawiające się wystąpienia obiektu będą się różnić z punktu widzenia upływu czasu choćby wartością atrybutu czasowego. Między tymi wystąpieniami obiektów mogą być zatem w bazie danych przechowywane dodatkowo związki dwojakiego rodzaju: bycia wystąpieniem obiektu i bycia wystąpieniem obiektu wyprowadzonym z innego wystąpienia. W ogólności wystąpienia obiektów tworzą graf skierowany, tzw. graf historii wystąpień, co przykładowo przedstawiono na schemacie 25.



Schemat 25. Związki wersji obiektów

(Źródło: Von Kim, 1994, s. 41)

Każde wystąpienie obiektu zawiera, oprócz wszystkich atrybutów określonych w definicji OBD (Obiektowej Bazy Danych) i przez użytkownika, dodatkowe trzy atrybuty nadawane przez system zarządzania obiektami:

- identyfikator wystąpienia (numer wystąpienia),
- identyfikator obiektu (rozumiany jako identyfikator klasy, obiektu w klasie),
- następnik wystąpienia (identyfikator obiektu następnika w grafie wystąpień obiektu).

Jeżeli następnik wystąpienia jest zerem (NULL), to obiekt jest reprezentantem zdarzenia bieżącego. Ze względu na jednorodność struktury i jednakowy algorytm obsługi obiektów, obiekty genetyczne (wystąpień obiektów OBD) są przechowywane w specjalnej strukturze również typu obiektowego.

Podjęcie systemowe potwierdza tezę, że czas istnieje niezależnie od zdarzeń i systemu, którego dotyczy. Innymi słowy zdarzenia są odwzorowaniem

stanów systemu lub jego otoczenia. Zdarzenia w odpowiedniej chwili docierają do bazy danych w formie komunikatów przechwytywanych przez moduł zarządzania bazami danych i obsługi komunikatów systemowych. Uzupełnieniem reprezentacji czasu jest utworzenie pewnych semantycznych zasad opisu zdarzeń w rachunkowości, mogących wspomagać proces projektowania i programowania obiektowych baz danych. Rachunkowość jest zbiorem zdarzeń silnie normowanych, które mogłyby podlegać logicznym sterowaniom wewnętrznym. Odpowiednia byłaby tu logika obowiązku, przyzwolenia i zabronienia, zwana w literaturze *logiką deodentyczną* (patrz: K. Kania, S. Kędzierski, J. Gołuchowski, 1997). Rozwiązania, z uwzględnieniem logiki deodentycznej, tworzyłyby podstawy większej naturalności w opisie rzeczywistości w językach programowania obiektowego. Przykładowo, ten sam towar sprzedany po terminie, jaki dopuszcza gwarancja, rodzi inne skutki prawne, roszczeniowe, stanu jakości (cena) itp. Podobnie jest w sprawozdawczości, wywiązywaniu się z umów zobowiązań i roszczeń należności. Problem formalnego opisu wraz klasyfikatorami przyzwolenia jest bardzo trudny, przede wszystkim w standaryzacji dokumentacji źródłowej), ale nie niemożliwy do realizacji. Integracja tych rozwiązań tworzy realne podstawy budowy baz inteligentnych, tj. samo kontrolujących się danych w systemie rachunkowości. Takim założeniom sprzyja oparcie się na paradygmacie obiektowym i traktowanie czasu — jako dodatkowego wymiaru — każdego zdarzenia odwzorowującego stany obiektów w bazie danych.

8.3. Organizacja systemu informatycznego

Przedstawione rozwiązania integracyjne rzeczywistego systemu rachunkowości i możliwych rozwinięć z reprezentacją czasu wymagają również innego spojrzenia na proces przetwarzania. Problem najważniejszy to duża ilość zapisów i konieczność dostarczania informacji w czasie rzeczywistym. Następnym problemem jest rozproszenie baz danych i konieczność współpracy z wieloma bazami równocześnie. Mimo dynamicznego rozwoju komputerów osobistych, powraca zapotrzebowanie na systemy komputerowe o większej elastyczności i mocy obliczeniowej z centralną bazą danych. Nie można tego organizować w oparciu o tradycyjne przetwarzanie na jednym serwerze, a tym bardziej o jednostanowiskowe komputery osobiste. Wytworzyła się sytuacja, w której wyraźnie oddziela się użytkowników systemu (klienci) od przetwarzania wykonawczego na specjalistycznym oprogramowaniu, obsługującym rozproszone bazy danych (serwer). Rozwiązania te noszą nazwę strategii (technologii)

klient/serwer. W tej technologii pojęcia „klient” i „serwer” należy traktować jako niematerialny proces, który dostarcza określoną funkcjonalność. Najczęściej mówi się o przejmowaniu przez serwery wszystkich specjalistycznych funkcji gromadzenia, przechowywania, udostępniania danych oraz ochrony danych. Programom użytkownika (klienta) pozostawia się inicjowanie przetwarzania i realizacje specjalistycznych algorytmów obsługi danych udostępnianych przez serwer z różnych źródeł, nawet bez wiedzy użytkownika. Zauważa się przy tym wyraźne tendencje w przejmowaniu wszystkich funkcji usługowych w programie przez systemy serwera.

Z technologią klient/serwer związane są następujące zagadnienia:

- obliczenia rozproszone; pojęcie to stosuje się, gdy więcej niż jeden niezależny proces odpowiada za wykonanie danego zadania, procesy te mogą być wywoływane na tych samych albo różnych platformach obliczeniowych,
- przetwarzanie rozproszone; oznacza, że wykonanie programu aplikacyjnego powoduje uruchomienie więcej niż jednego niezależnego procesu; o przetwarzaniu rozproszonym mówi się tylko wtedy, gdy pojedynczy etap pracy jest wykonywany przez wiele procesów,
- rozproszone bazy danych; dane zawarte w rozproszonych bazach danych podzielone są między kilka baz, ale dla aplikacji stanowią logiczną, pojedynczą bazę danych.

Termin klient/serwer związany jest z relacją między dwoma systemami albo procesami. W relacji tej *klient* jest systemem, który zleca jakieś zadanie systemowi zwanemu *serwerem*⁷⁷. Często również program użytkowy, który inicjuje komunikację z innym programem, określane jest jako program klient (program kliencki) lub klient. Natomiast program, który oczekuje na zgłoszenia od klientów, czyli na żądania nawiązania komunikacji, określane jest terminem serwer⁷⁸. Klient jest zazwyczaj wywoływany przez użytkownika, który pragnie skorzystać z usługi oferowanej w sieci. Program taki rozpoczyna działanie od nawiązania kontaktu z serwerem. Następnie wysyła do serwera zapytanie (tzn. żądanie wykonania usługi) i czeka na odpowiedź. Po otrzymaniu odpowiedzi od serwera kontynuuje działanie. Programy klienckie są najczęściej łatwiejsze do skonstruowania niż serwery, ponieważ program klient nie musi uzyskiwać specjalnych przywilejów systemowych, aby móc wykonywać swoje zadania. Serwery natomiast muszą mieć dostęp do zasobów chronionych przez system operacyjny, takich jak dane i procedury systemowe lub też zarezerwowane porty protokołów komunikacyjnych. Dlatego też muszą działać w trybie uprzywilejowanym, z uprawnieniami programu systemowego⁷⁹.

W modelu przetwarzania typu klient/serwer użytkownicy pracują na inteligentnych komputerach, które nazywa się systemami czołowymi (ang. *front-end*) i które komunikują się z serwerem zaplecza (ang. *back-end*), realizującym różnorodne usługi, między innymi dostęp do bazy danych, zarządzanie siecią, a także centralne składowanie plików⁸⁰. W przetwarzaniu typu klient/serwer sieć komputerowa pełni rolę platformy komunikacyjnej, na bazie której wielu użytkowników korzysta z usług jednego lub większej liczby serwerów. Wzajemne oddziaływanie między aplikacją użytkownika a programem (bazą danych lub sieciowym systemem operacyjnym), który działa na serwerze back-end nazywa się relacją klient/serwer (ang. *client-server relationship*). Klient korzysta z komputera, który dysponuje własną mocą obliczeniową, na którym działa program obsługujący komunikację z użytkownikiem. Wynika z tego, że przetwarzanie w modelu klient/serwer zastępuje klasyczny model przetwarzania, gdzie program użytkownika rozstrzygał równocześnie o metodach dostępu do danych jak i ich przetwarzaniu.

Model przetwarzania typu klient/serwer odnosi się do systemów operacyjnych oraz do aplikacji. W relacji klient/serwer przetwarzanie jest rozdzielone pomiędzy system klienta i serwera. W systemie klienta działa aplikacja odpowiedzialna za wyświetlanie interfejsu użytkownika. Aplikacja ta odpowiednio formatuje zapytania dotyczące usług sieciowych i wyświetla informacje lub komunikaty odebrane z serwera. Serwer realizuje operacje przetwarzania typu back-end, takie jak sortowanie danych lub sporządzanie wyciągów. Po zakończeniu sortowania, sporządzania wyciągu lub wykonywania innej czynności zleconej przez użytkownika, serwer odsyła klientowi wyniki. Powoduje to ograniczenie ruchu w sieci, ponieważ klient otrzymuje tylko żądane informacje, a nie duże bloki danych, z których musiałby wybrać te, które go interesują. Rolę serwerów w środowiskach klient/serwer pełnią głównie systemy typu superserwer, minikomputery lub systemy mainframe, które są w stanie obsłużyć jednocześnie zapytania dużej liczby użytkowników, a przy tym realizować zapytania związane z bezpieczeństwem i zarządzaniem siecią.

Technologia klient/serwer to sposób na budowanie niedrogich środowisk przetwarzania, które charakteryzują się łatwością dostosowania do specyficznych aplikacji. Jeśli chodzi o oprogramowanie, które działa w systemie klient/serwer, jest to zazwyczaj system zarządzania bazą danych DBMS (ang. *Database Management System*), w którym klienci posługują się językiem zwanym strukturalnym językiem zapytań SQL (ang. *Structured Query Language*), formułując zapytania do serwerów back-end. Systemy bieżącego przetwarzania

transakcji OLTP (ang. *Online Transaction Processing*) są szczególnie dobrze dostosowane do modelu klient/serwer. Jednak najczęściej spotyka się serwery plików i serwery baz danych, natomiast systemy serwerów back-end mogą równie dobrze świadczyć usługi związane z połączeniami dedykowanymi lub drukowaniem⁸¹.

Zaprezentowane poniżej cechy architektury klient/serwer są wskazówkami i podstawą do budowy i projektowania różnie połączonych systemów, a mianowicie:

- A. *Blok klienta* — jest to zestaw procesów, które realizowane są po stronie klienta. Wykonuje on usługi systemu operacyjnego. System operacyjny natomiast dostarcza graficzny łącznik z użytkownikiem GUI (ang. *Graphical User Interface*) oraz mechanizm komunikacji z procesami, które są realizowane na innych maszynach.
- B. *Blok warstwy pośredniej* — wykonuje się po stronie klienta i po stronie serwera lub na maszynie, która jest wydzielona i obsługuje warstwę pośrednią. Blok ten podzielono na kilka warstw, z których mogą korzystać bloki serwera i klienta. Celem tego bloku jest tworzenie luźno połączonych systemów.
- C. *Blok serwera* — to zestaw procesów realizowanych po stronie serwera. Klient, tj. komputer PC, wysyła zapytanie odczytu porcji pliku przez sieć. Natomiast funkcjonalność przetwarzania danych, które przechowywane są po stronie serwera, zostaje wykonana przez klienta. Efektem tych działań jest duża wymiana komunikatów przez sieć oraz minimalne izolacje procesów serwera i klienta. Blok serwera realizuje utrzymanie baz danych oraz związane z nimi składnice danych. Składnica danych jest bazą danych, która realizuje przetwarzanie analityczne stanowi podstawę dla systemów wspomagania decyzji. System wspomagania decyzji dostarcza informacji użytkownikowi. Dzięki tym informacjom może on analizować sytuację i podejmować decyzje. Natomiast systemy przetwarzania analitycznego wspierają proces decyzji strategicznych i taktycznych. Użyte w nich dane mają charakter historyczny. Są traktowane jako dane tylko do odczytu, za ich pomocą można dokonywać obserwacji faktów z przeszłości i prognozować w przyszłość.

Składnicę danych projektuje się dla dużych ilości danych przeznaczonych tylko do odczytu. Architektura składnicy danych jest zestawem reguł i struktur, które są podstawą projektowania wszystkich systemów i produktów danego rodzaju.

Wyszczególnia się reguły, które określają architekturę składnic danych:

- źródłem danych do składnicy danych są systemy, które przechowują dane w różnych systemach baz danych lub plikach, na różnych serwerach,
- dane z systemów źródłowych są integrowane i zmieniane przed ładowaniem do składnicy danych — w przypadku danych, które pochodzą z różnych systemów należy dopasować je do funkcji, które są określone dla składnicy danych,
- składnica danych jest wydzieloną bazą danych tylko do odczytu, jej zadanie polega na wspieraniu podejmowania decyzji — oznacza to, że składnica danych nie może pracować na danych operacyjnych, a dane mają charakter historyczny,
- aby mieć dostęp do składnicy danych, korzysta się z architektury klient/serwer, serwerem w tym przypadku jest baza danych ze składnicą danych, klientem natomiast jest program lub narzędzie graficzne — umożliwiające wykonywanie przygotowywanych raportów, a także tworzenie własnych modeli analizy tych danych.

Obsługę rozproszonych obiektów zapewnia trójwarstwowa architektura systemów. Między procesem klienta i serwera jest dodana warstwa pośrednia. Dostarcza ona usługi komunikacji oraz realizuje usługi strategii (przedstawione w postaci logicznych obiektów biznesu, rozmieszczone dowolnie w sieci lokalnej i rozległej). Założenia technologii obiektów rozproszonych są podobne do założeń architektury klient/serwer. Celem jej jest rozszerzenie funkcjonalności i uporządkowanie rozwoju systemów informatycznych, aby mogły one sprostać wymaganiom, które stawiane są przez systemy zarządzania organizacjami. Obiekt rozproszony to niezależny zestaw kodu, który jest zlokalizowany w dowolnym miejscu w sieci. Do zestawu tego mogą mieć dostęp klienci poprzez określony zestaw łączników (można korzystać z usług zdefiniowanych przez konstruktora obiektu). Obiekty rozproszone są umieszczone w warstwie pośredniej logicznie, inaczej warstwie strategii, która modeluje obiekty świata rzeczywistego — obiekty biznesu.

Warstwa strategii ma na celu:

- prezentowanie obiektów świata rzeczywistego w języku użytkownika,
- zapewnienie hermetyzacji operacji, które są realizowane wewnątrz obiektu,
- kojarzenie technologii dostawców po stronie klienta, a także po stronie serwera,
- dostarczenie komputerów w postaci obiektów biznesu; mogą one być łączone w celu dopasowania do określonych potrzeb odbiorcy.

- obsługę systemów pracy grupowej, np. serwery poczty elektronicznej, elektronicznych dokumentów EDI, automatyzacji obiegu procesów pracy, harmonogramowanie.

Istnieje kilka wariantów konfiguracji środowiska klient/serwer. W pierwszym rozwiązaniu kilku klientów korzysta z jednego serwera; jest to typowa konfiguracja, którą spotyka się w niewielkich sieciach lokalnych (LAN). Natomiast w sieciach typu *peer-to-peer*, takich jak np. Microsoft Windows for Workgroups, stacje robocze mogą pełnić funkcje zarówno klientów, jak i serwerów. Użytkownik może udostępnić pliki ze swojego dysku innym użytkownikom sieciowym. W ten sposób stacja robocza zaczyna pełnić funkcję serwera, który działa na rzecz innych klientów. Jednocześnie ten sam użytkownik może korzystać jako klient z plików na innych stacjach roboczych.

W większości konfiguracji komunikacja realizowana jest za pośrednictwem sieci lokalnych (LAN). Serwery mogą należeć do poszczególnych oddziałów lub grup roboczych lub mogą być zlokalizowane centralnie i dostępne dla całej instytucji. Serwery dostępne centralnie nazywane są serwerami korporacyjnymi. Serwery mogą również znajdować się w odległych miejscach, dostępnych dla użytkowników za pośrednictwem łączy telekomunikacyjnych. Projektanci i zarządcy sieci powinni wziąć pod uwagę wydłużony czas odpowiedzi w komunikacji z odległymi serwerami. Czas ten zależy od rodzaju połączenia. Tam gdzie nie jest wymagany dostęp w czasie rzeczywistym, można zastosować system przesyłania komunikatów — użytkownicy przesyłają wtedy zapytania do serwera, a serwer przekazuje odpowiedzi w formie wiadomości (komunikatu), umieszczonej w skrzynce elektronicznej (*ang. mailbox*) użytkownika. Odpowiedź taka może nadejść za kilka sekund, minut, a nawet godzin, w zależności od rodzaju połączenia i warunków czasowych narzuconych przez projektantów systemu.

Korzyści, które wynikają ze stosowania architektury klient/serwer są następujące:

- przetwarzanie w modelu klient/serwer ułatwia instytucjom łagodne przejście od środowisk opartych na systemach *mainframe* i mikrokomputerach do rozwiązań, w których wykorzystuje się serwery sieci lokalnych i stacje robocze; sieci komputerowe pełnią funkcję platformy komunikacyjnej o zasięgu korporacyjnym,
- obciążenie zadaniami, pochodzące od oprogramowania aplikacyjnego, rozłożone jest na wiele systemów komputerowych; komputery klientów samo-

dzielnie wykonują dużą część swoich zadań, natomiast przetwarzanie rozłożone jest na wiele systemów mikrokomputerowych,

- serwery, wykonując czynności związane z obsługą centralnych danych korzystają z bezpośrednio przyłączonej pamięci masowej, w której dane te są przechowywane, co powoduje zmniejszenie się ilości informacji przesyłanych w sieci; duża część informacji przepisywana jest jednorazowo do pamięci podręcznej serwera, a nie do pamięci poszczególnych stacji roboczych, które z tych informacji korzystają,
- zmniejsza się ruch w sieci, gdyż serwer przekazuje klientowi jedynie żadaną informację, a nie duże bloki danych, które wymagałyby dalszego przetwarzania w stacji roboczej,
- duże systemy, które pełnią funkcję serwerów, nie muszą obsługiwać tych aplikacji, które lepiej nadają się do wykonywania na stacjach roboczych,
- zapewnione jest bezpieczeństwo danych zgromadzonych w jednym miejscu; przechowywanie informacji w składnicach danych pozwala na udostępnianie części danych serwerom pośredniczącym bez utraty centralnej kontroli nad danymi,
- dzięki scentralizowaniu danych administratorzy mogą korzystać z systemów kontroli bezpieczeństwa, wprowadzających ograniczenia w dostępie do danych oraz stosować mechanizmy monitorowania dostępu.

Mechanizmy pracy w modelu klient/serwer otwierają możliwości równoległego przetwarzania danych przez większą liczbę systemów; wiele komputerów wspólnie pracuje nad realizacją pojedynczego zadania związanego z przetwarzaniem danych. Każdemu z systemów przydzielana jest część zadania, a uzyskane wyniki są łączone tak, że zadanie wykonane jest o wiele szybciej niż byłoby to możliwe przy użyciu pojedynczego systemu komputerowego⁸². Technologia klient/serwer jest popularna, ponieważ: umożliwia zastosowanie taniego sprzętu, zapewnia skalowalność, jest tolerancyjna na błędy, umożliwia łatwe zarządzanie rozproszeniem danych, umożliwia uruchomienie na stacji roboczej interfejsu graficznego użytkownika. Technologie serwera plików preferują małe i duże firmy, ponieważ koszty instalacji i utrzymania tego rozwiązania są niskie. Korzysta się najczęściej z rozwiązań oprogramowania pracującego na prostych komputerach klasy PC.

W podsumowaniu wizji przyszłościowego systemu informatycznego rachunkowości można podkreślić, że dominuje tu dążenie do pełnej obsługi stanowiska pracy księgowego, ekonomisty a w mniejszych firmach nawet właściciela. Rachunkowość oparta transakcjach, tj. faktycznych dowodach finanso-

wo-rzeczowych, jest źródłem tworzenia bazy danych typu historycznego (właściwie: składnicy danych) o możliwych rozwinięciach temporalno-deodentycznych. Na przedstawione rozwiązania własne i w oparciu o omówioną literaturę nakłada się chęć budowy systemu o cechach układu cybernetycznego. Kształt takiego systemu został sformułowany w kolejnych rozdziałach tej pracy. Główne opcje systemu informatycznego to:

- pełne ujęcie zdarzeń firmy (transakcji, czynności) wraz z ich klasyfikatorami w komputerze,
- zamknięcie cyklu przetwarzania informacji w postaci procesu informacyjno-decyzyjnego (baza faktów – identyfikacja i analiza gospodarności, prognozy finansowe – optymalizacja budżetu).

Na te problemy nakłada się realizacja aktywnych powiązań sieciowych (INTERNET, POLPAK, etc.) z kontrahentami i bankami, np. przy pomocy rozwiązań klasy HOME BANK. Dla rachunkowości w dalszym ciągu dyskusyjne pozostaje: czy w budowie serwera danych opieramy się na jednostkowych dowodach źródłowych, czy posługujemy się zintegrowanym dowodem zdarzenia gospodarczego, co przedstawiono w modelu obiektowym. W pierwszym przypadku baza danych (obiektoowo-temporalno-deodentyczna) lepiej opisywałaby rzeczywistość, ale trudniejsza byłaby do zaprojektowania i oprogramowania. W drugim natomiast kompleksowo można rozpatrywać problemy modelowania stanu kont i zasobów rzeczowych w przedsiębiorstwie. Budowa takiego systemu w architekturze klient/serwer nie jest już niczym nieosiągalnym. Możliwości takie daje między innymi pakiet VISUAL OBJECT (firmy Computer Associates) akceptujący kod źródłowy, w którym opracowano dyskutowany system rachunkowości transakcyjnej TRANS.

PRZYPISY

1. P. Sienkiewicz, *Inżynieria systemów*. 1988, s. 13.
2. J. Kisielnicki, *Informatyczna infrastruktura zarządzania*. Warszawa 1993, s. 48.
3. B. Nogalski, T. Biełas, M. Czapiewski, *Zarządzanie w różnych formach własności*. Gdańsk 1994, s. 9.
4. U. Gross, *Zarządzanie marketingowe*, w: *Zarządzanie małą firmą*, red. H. Bieniok. Katowice 1995.
5. Z. Dowgiałło (red.), *Słownik ekonomiczny dla przedsiębiorcy*. Szczecin 1996, s. 226.
6. B. Klimczak, *Strategie przedsiębiorstw w świetle współczesnych teorii mikroekonomicznych*. W: *Systemy informatyczne w zarządzaniu strategicznym*. KI PAN Oddział w Gdańsku. 1998, s. 53.
7. R. Budziński, *Komputerowy system rachunkowości rolnej w technologii SBD*. Warszawa 1991.
8. M.J. Earl, *Management Strategies for Information Technology*. New York 1989.
9. G.A. Steiner, J.B. Meiner, *Management Policy and Strategy*. New York 1986, s.5; a także: K. Obłój, M. Trybuchowski, *Znaczenie, elementy i typy strategii*, w: *Zarządzanie - teoria i praktyka*, red. A. Koźmiński, W. Piotrowski. Warszawa 1995, s. 123.
10. A. Stabryła, *Zrządzanie rozwojem firmy*. Kraków 1996, s. 21.
11. A. Hax, N. Majluf, *Strategic Managment. An Intergravite Perspective*. New York 1984.
12. J. Stoner, Ch. Wankel, *Kierowanie*. Warszawa 1992, s. 95; oraz: W. Kieżun, *Sprawne zarządzanie organizacją*. Warszawa 1997.
13. K. Obłój, M. Trybuchowski, *Znaczenie...*, dz. cyt., s. 123.
14. M.E. Porter, *Competitive Strategy. Techniques for Analyzing Industries and Competotors*. New York 1985.
15. A. Stabryła, *Zarządzanie...*, dz. cyt., s. 26.
16. R. Hartley, *Marketing Fundamentals*. Harper & Row, New York 1983; oraz: E.J. McCarty, S.J. Shapiro, W.D. Perreault, *Basic Marketing, A Managerial Approach*. Irwin, Boston 1989; a także: U. Gross, dz. cyt., s. 135-137.

17. A. Stabryła, *Zarządzanie...*, dz. cyt., s. 28.
18. U. Gross, *Zarządzanie...*, dz. cyt., s. 139.
19. Altkorn J., *Podstawy marketingu*. Kraków 1995, s. 396.
20. K. Obłój, M. Trybuchowski, *Znaczenie...*, dz. cyt., s. 136.
21. Cz. Sikorski, *Filozofia zarządzania nowoczesnym przedsiębiorstwem*. Warszawa 1995, s. 7.
22. B. Kubiak, *HUMAN-COMPUTER INTERAKTION*. Gdańsk 1997, s. 9.
23. W.P. Beck, *Corporate Planing for an Uncertain Future*. 1982.; oraz J.K. Galbraith, *Ekonomia w perspektywie - krytyka historyczna*. Warszawa 1991.
24. *Podstawy rachunkowości*, pod. red. K. Sawickiego. Warszawa 1996, s. 13.
25. T. Kiziukiewicz, *Rachunkowość. Zasady prowadzenia w jednostkach gospodarczych*. Wrocław 1995, s. 12.
26. M. Klimas, Z. Messner, *Teoretyczne podstawy rachunkowości*. Warszawa 1986, s.10.
27. A. Jaruga, I. Sobańska, L. Kopczyńska, A. Szychta, E. Walińska, *Rachunkowość dla menadżrów*. Łódź 1995, s. 7.
28. M. Dobija, *Rachunkowość zarządcza*. Warszawa 1995, s. 45.
29. E. Nowak, *Rachunkowość jako system informacyjny przedsiębiorstwa*. „Prace Naukowe Akademii Ekonomicznej we Wrocławiu” nr 690, 1994.
30. T. Ciesielczyk, G. Watras, *Rachunkowość a systemy wspomagania decyzji*. 1994, Wrocław.
31. *Rachunkowość - system informacyjny controllingu*, pod. red. E. Nowaka. Wrocław 1993.
32. T. Ciesielczyk, G. Watras, *Rachunkowość...*, dz. cyt.
33. M. Klimas, Z. Messner, *Teoretyczne...*, dz. cyt..
34. D. Misińska, *Podstawy rachunkowości*. Warszawa 1994, s. 11.
35. A. Jaruga, I. Sobańska, L. Kopczyńska, A. Szychta, E. Walińska, *Rachunkowość...*, dz. cyt., s. 8.
36. J. Aleszczyk, *Rachunkowość od podstaw*. Poznań 1995, s. 13-14.
37. T. Kiziukiewicz, *Rachunkowość...*, dz. cyt., s. 12.
38. J. Aleszczyk, *Rachunkowość...*, dz. cyt., s. 14.
39. M. Klimas, Z. Messner, *Teoretyczne...*, dz. cyt., s. 14.
40. *Podstawy...*, dz. cyt., s. 15.

41. *Podstawy...*, dz. cyt., s. 15.
42. B.F. Kubiak, A. Korwicki, *Restrukturyzacja zarządzania procesami gospodarczymi współczesnej organizacji z wykorzystywaniem technologii informacji*. Gdańsk 1997, s. 29.
43. T. Kiziukiewicz, *Rachunkowość...*, dz. cyt., s. 13.
44. A. Jaruga, I. Sobańska, L. Kopczyńska, A. Szychta, E. Walińska, *Rachunkowość...*, dz. cyt., s. 8.
45. P. Sienkiewicz, *Inżynieria systemów*. 1988, s. 41.
46. K. Kania, J. Gołuchowski, *Zagadnienia czasu w klasycznych systemach baz danych*. „Informatyka” 1996, nr 7, s. 6.
47. K. Kania, J. Gołuchowski, *Zagadnienia...*, dz. cyt., s. 5.
48. R. Budziński, B. Śmiałkowska, *Reprezentacja czasu w systemach baz finansów przedsiębiorstwa*. Gdańsk 1997.
49. T. Wierzbicki, *System informacji gospodarczej*. Warszawa 1981, s. 10.
50. I. Dziedziczak, *Organizacja bazy danych księgowych*. Warszawa 1983, s. 102.
51. G. Sorter, *An „Events” Approach to Basic Accounting Theory*. „The Accounting Review” 1969 no. 9.
52. I. Dziedziczak, *Organizacja...*, dz. cyt.
53. A. Zaleski, *Ustawa o rachunkowości a komputery*. 1994, s. 172.
54. Dziedziczak, *Organizacja...*, dz. cyt., s. 114.
55. W. Gos, *Sposoby doskonalenia informacyjnej funkcji rachunkowości*. Szczecin 1994.
56. W. Harris, *Bazy Danych (nie tylko dla ludzi biznesu)*. Warszawa 1994, s.175.
57. T. Nelke, . 1988, s. 82.
58. Aleszczyk J., *Rachunkowość...*, dz. cyt., s. 6.
59. *Teoretyczne podstawy rachunkowości*, pod. red. T. Peche. Warszawa 1988, s. 92-103.
60. M. Klimas, Z. Messner, *Teoretyczne...*, dz. cyt., s. 40.
61. Aleszczyk J., *Rachunkowość...*, dz. cyt., s. 167-172.
62. R. Budziński, *System informatyczny obsługi naczelnego kierownictwa (SNK) - synteza rozwiązań na przykładzie administracji państwowej*. „Zeszyty Naukowe Uniwersytetu Szczecińskiego” nr 164, 1994, s. 193.

63. H. Jezierska, *Słownik Informatyki*. Warszawa 1989, s. 83.
64. P. Sienkiewicz, *Inżynieria...*, dz. cyt., s. 107.
65. K. Kania, J. Gołuchowski, *Systemy temporalnych baz danych*. 1996, s. 23.
66. C. Hall, *Techniczne podstawy systemów klient-serwer*. Warszawa 1996, s. 23.
67. D.E.R. Denning, *Kryptografia i ochrona danych*. Warszawa 1991, s. 16.
68. B.F. Kubiak, A. Korwicki, *Restrukturyzacja...*, dz. cyt., s. 29.
69. M. Gruber, *SQL*. Warszawa 1996, s. 15.
70. S. Węgrzyn, *O kierunkach rozwoju i o polityce naukowej w obszarze informatyki*, Referat wygłoszony na seminarium Sekcji Informatyki KBN w Zakopanem. 1996, s. 12.
71. D.A. Taylor, *Technika obiektowa*. Warszawa 1994, s. 20.
72. T. Kiziukiewicz, *Rachunkowość...*, dz. cyt., s. 12.
73. *Podstawy...*, dz. cyt., s. 14.
74. Von Kim, *Wprowadzenie do obiektowych baz danych*. Warszawa 1996, s. 28.
75. R. Budziński, B. Śmiałkowska, *Reprezentacja...*, dz. cyt., s. 64.
76. Von Kim, *Wprowadzenie...*, dz. cyt., s. 167.
77. C. Hall, *Techniczne...*, dz. cyt., s. 23.
78. D. Comer, D. Stevens, *Sieci komputerowe TCP/IP. Programowanie w trybie klient-serwer-wersja BSD*. Warszawa 1997, s. 37.
79. D. Comer, D. Stevens, *Sieci...*, dz. cyt., s. 37.
80. T. Sheldon, *Wielka Encyklopedia Sieci Komputerowych*. Wrocław 1995, s. 184.
81. T. Sheldon, *Wielka...*, dz. cyt., s. 185-186.
82. T. Sheldon, *Wielka...*, dz. cyt., s. 190-191.

LITERATURA

- Aleszczyk J., *Rachunkowość od podstaw*. Poznań 1995.
- Altkorn J., *Podstawy marketingu*. Kraków 1995.
- Budziński R., *Komputerowy system rachunkowości rolnej w technologii SBD*. Warszawa 1991.
- Budziński R., *Practical solutions of an information system of data bases. Applications of Computer Systems ACS*. Szczecin 1994.
- Budziński R., *Rachunkowość transakcyjna*. Warszawa 1994.
- Budziński R., *System analizowania i prognozowania procesów gospodarczych z uwzględnieniem problemów z zakresu ochrony środowiska w regionie szczecińskim*. Szczecin 1994.
- Budziński R., *System informatyczny obsługi naczelnego kierownictwa (SNK) - synteza rozwiązań na przykładzie administracji państwowej*. „Zeszyty Naukowe Uniwersytetu Szczecińskiego” nr 164, 1994.
- Budziński R., Głodek Z., *Projektowanie edycji i język użytkownika w systemach baz finansów przedsiębiorstwa*. „Studia Informatica US” (praca w druku, 1997).
- Budziński R., Gos W., Nadolna B., *Komputerowy model rachunkowości transakcyjnej (ewidencja finansowa)*. „Zeszyty Teoretyczne Rady Naukowej Stowarzyszenia Księgowych w Polsce” nr 19, 1991.
- Budziński R., Śmiałkowska B., *Reprezentacja czasu w systemach baz finansów przedsiębiorstwa*. Gdańsk 1997.
- Comer D., Stevens D., *Sieci komputerowe TCP/IP. Programowanie w trybie klient-serwer-wersja BSD*. Warszawa 1997.
- Denning D.E.R., *Kryptografia i ochrona danych*. Warszawa 1991.
- Douglas K.B., *The object database handbook*. Wiley Computer Publishing. New York Chichester - Brisbane - Toronto - Singapore 1996.
- Dziedziczak I., *Organizacja bazy danych księgowych*. Warszawa 1983.
- Gruber M., *SQL*. Warszawa 1996.
- Gupta R., Horowitz E., *Object-oriented database with applications to CASE, networks and VLSI CAD*. Prentice Hall series in Data and Knowledge Base System. Englewood Cliffs, New Jersey 1991.
- Hall C., *Techniczne podstawy systemów klient-serwer*. Warszawa 1996.

- Harris W., *Bazy Danych (nie tylko dla ludzi biznesu)*. Warszawa 1994.
- Hartley R., *Marketing Fundamentals*. Harper & Row, New York 1983.
- Jeziarska H., *Słownik Informatyki*. Warszawa 1989.
- Kania K., Gołuchowski J., *Reprezentacja czasu w modelu konceptualnym SI*. „Informatyka” 1996, nr 8.
- Kania K., Gołuchowski J., *Systemy temporalnych baz danych*. „Informatyka” 1996, nr 9.
- Kania K., Gołuchowski J., *Zagadnienia czasu w klasycznych systemach baz danych*. „Informatyka” 1996, nr 7.
- Kania K., Kędziński St., Gołuchowski J., *Zależności temporalne w modelowaniu i analizie procesów gospodarczych*. „Informatyka” 1998, nr 3/98, s. 42.
- Kisielnicki J., *Informatyczna infrastruktura zarządzania*. Warszawa 1993.
- Kiziukiewicz T., *Rachunkowość. Zasady prowadzenia w jednostkach gospodarczych*. Wrocław 1995.
- Klimas M., Messner Z., *Teoretyczne podstawy rachunkowości*. Warszawa 1986.
- Klimczak B. (1997), *Strategie przedsiębiorstw w świetle współczesnych teorii mikroekonomicznych*. W: Systemy informatyczne w zarządzaniu strategicznym. KIPAN Oddział w Gdańsku.
- Lange O., *Cybernetyka a ekonomia*. „Biuletyn Polskiego i Towarzystwa Cybernetycznego” nr 1, 1965.
- Ling D., Bell D., *Modelling and Managing Time in Database Systems*. „The Computer Journal” vol. 35, no. 4, 1990.
- Ling D., Bell D., *Taxonomy of Time Models in Databases*. „Information and Software Technology” vol.32, no. 3, 1990.
- Llewellyn M., Bassiouni M., *Historical Database Views*. „Information and Software Technology” vol. 33, no. 2, 1991.
- Maiocchi R., Percini B., *Temporal Data Management Systems: A Comparative View*. „IEEE Transaction on Knowledge and Data Engineering” vol. 3, no. 4, 1991.
- Matuszewicz J., *Rachunkowość przedsiębiorstw i instytucji*. Warszawa 1987.
- McCarty E.I., Shapiro S.I., Perreault W.D., *Basic Marketing, A Managerial Approach*. Irwin, Boston 1989.
- Mykowiecki T., *DBASE, FOXPRO, Bazy danych*. Warszawa 1992.

- Nogalski B., Biełas T., Czapiewski M., *Zarządzanie w różnych formach własności*. Gdańsk 1994.
- Nowak E., *Rachunkowość jako system informacyjny przedsiębiorstwa*. „Prace Naukowe Akademii Ekonomicznej we Wrocławiu” nr 690, 1994.
- O’Shaughnessy J., *Organizacja i zarządzanie w przedsiębiorstwie*. Warszawa 1972.
- Penc J., *Strategie zarządzania. Perspektywiczne myślenie, systemy działania*. Warszawa 1994.
- Podstawy rachunkowości*, pod. red. K. Sawickiego. Warszawa 1996.
- Rachunkowość - system informacyjny controllingu*, pod. red. E. Nowaka. Wrocław 1993.
- Roddick J.F., *SQL/SE a Query Language Extention for Databases Supporting Schema Evolution*. „Sigmond Record” vol. 21, no. 3, 1992.
- Sheldon T., *Wielka Encyklopedia Sieci Komputerowych*. Wrocław 1995.
- Sienkiewicz P., *Inżynieria systemów*. Warszawa 1991.
- Snodgrass R., *The Temporal Query Language TQUEL*. „ACM TODS” vol. 12 no. 2, 1987.
- Snodgrass R., Ahn I., *Temporal Databases*. „Computer” 1986 no. 19.
- Sorter G., *An „Events” Approach to Basic Accounting Theory*. „The Accounting Review” 1969 no. 9.
- System naczelnego kierownictwa w zarządzaniu*, pod. red. R. Budzińskiego. Szczecin 1997.
- Tansel A., *Modelling Temporal Data*. „Information and Software Technology” vol. 32, no. 8, 1990.
- Tansel A., Erol Arkun M., Ozsoyoglu G., *Time-by-Example Query Language for Historical Databases*. „IEEE Transction on Software Engineering” vol. 15, no. 4, 1989.
- Taylor D. A., *Technika obiektowa*. Warszawa 1994.
- Teoretyczne podstawy rachunkowości*, pod. red. T. Peche. Warszawa 1988.
- Theodoulidis C., Loucopoulos P., *The time dimension in conceptual modeling*. „Information Systems” vol. 16, no. 3, 1991.
- Thompson A.A., Stricland J., *Strategic management Concepts and bases*. Homewood Illinois 1987.

-
- Ustawa o rachunkowości z dnia 29.09.1994 r.* „Dziennik Ustaw” 1994 nr 121.
- Wang X., S., Bettini C., Brodski A., Jajodia S., *Logical Design for Temporal Databases with Multiple Granularities*. „ACM Transactions on Database Systems” vol. 22, no. 2, 1997.
- Węgrzyn S., *O kierunkach rozwoju i o polityce naukowej w obszarze informatyki*, Referat wygłoszony na seminarium Sekcji Informatyki KBN w Zakopanem. 1996.
- Wierzbicki T., *System informacji gospodarczej*. Warszawa 1981.
- Von Kim, *Wprowadzenie do obiektowych baz danych*. Warszawa 1996.
- Zaleski A., *Ustawa o rachunkowości a komputery*. „Rachunkowość”. Zeszyt Specjalny 1994.

SYSTEM OF TRANSACTIONAL ACCOUNTING

(information system, algorithms, models)

SUMMARY

The work presents a set of author solutions concerning time representation in accounting databases. The inspiration for this work was a great number of unsatisfied users using particular information systems, i.e. systems based on a model of the result directory. The reason is the conflict between solutions of formal accounting and the need of information. The business goes, when there is an occasion and clients, but formal information flows monthly or yearly.

The solutions presented in the work are original proposals, particularly automatic accounting, algorithms of informing editor and query language of a user. They appease inconveniences existing in that kind of management information systems. The base for discussed system functionality is the description of events/transactions with daily accuracy. Therefore the data can be organised according to {year:month:day} configurations. Additionally the certain vision of generic integration in accounting in the field of relational and object data bases is presented.

The object model of accounting presented in the work shows profits of economical events storage at the level of smaller entity (parts). One of the most important features in the context of object databases (inheritance and multi-version) lets to create infinite number of solutions transitional (and virtual) and information services about the state of a firm. They are supported by the same data but from different levels of aggregation. It is important, that any change in values of source events (entities) is automatically seen in all system developments. The border between the system software and database disappears, this is the weakness of data determinism. It concerns mainly resulting data, which can be modified temporary or constantly modelled by object databases languages.

Rachunkowość transakcyjna nie ma, jak dotychczas, swej precyzyjnej definicji literaturowej. Najogólniej można przyjąć, że chodzi tu o oparcie ewidencji na aktywnych transakcjach gospodarczych, czyli na przechowywaniu wszystkich danych - dokumentów źródłowych wraz z ich klasyfikatorami - w komputerze. Ze zbioru tego mogą być, na różne przedziały czasu, rozwijane serwisy informacji i modelowane stany finansów przedsiębiorstwa. Uzasadnia to użycie sformułowania „aktywnych transakcji”, które utożsamia się z możliwością logicznego manipulowania danymi opisującymi zdarzenia gospodarcze. Podstawą jest założenie (co jest oczywiste), że prowadzimy działalność gospodarczą przede wszystkim poprzez zawierane transakcje. Znajomość opłacalności zawieranych kontraktów (transakcji) jest podstawą sprawnego myślenia i działania ekonomicznego. Nowym ujęciem jest również reprezentacja czasu. Problem ten będzie, w niedalekiej przyszłości, uważany za najważniejszy do rozwiązania w informatycznych systemach zarządzania. Czas jest stanem natury, bez którego nie istnieje pojęcie przeszłości, rzeczywistości czy przyszłości. Nie ma zatem mowy o postępie, czy odwoływania się do doświadczeń z przeszłości w klasycznym (migawkowym) ujmowaniu baz danych.

- **Rozwiązania podkatalogowe**
- **Symulacja komputerowa**
- **Systemy wewnętrzne**

MODELOWANIE

ISBN 83-85847-23-5
ISSN 0208-8029