

Obiektowe podejście i parametryzacja obiektów jako podstawa rozwoju produktów bankowych ¹

1. Nawiązanie do metod obiektowego podejścia

Wydaje się, że metody te niedostatecznie podkreślają dynamiczne aspekty obiektu, co być może wywodzi się jeszcze z podejścia ER (Entity Relationship) zaproponowanego przez Petera Chena w 1976 roku. Krok pewien w kierunku odwzorowania dynamicznego stanowił model informacyjny Shlaera i Mellora (1988 r), w którym obiekt charakteryzowany (czy też identyfikowany) jest przez rolę jaką odgrywa, przez zdarzenia pojawiające się w określonym momencie czasu czy też przez tzw. interakcje w jakie wchodzi. W metodzie OMT (Object Modeling Technique) Rumbaugh docenione zostaje zachowanie się obiektu określane jako operacja, w szczególności zaś tzw. operacji polimorficznej znajdującej zastosowanie w stosunku do obiektów należących do różnych klas.² Inną zaletą Rumbaugh jest oderwanie się od technologii relacyjnych baz danych dzięki zaprzestaniu używania tzw. kluczy pierwotnych i obcych (primary and foreign keys).

Sądzę, że mimo 25 letniej historii strukturalnego i obiektowego podejścia, wiele jeszcze pozostaje do dokonania w kierunku dynamicznego modelowania biznesowych obiektów, opartego na symulacji działalności przedsiębiorstwa i posiadającego transparentne przełożenie na komputerową technologię realizacji. Wymaga to zespolenia różnorodnych metod, narzędzi i technologii, w tym języków symulacyjno-specyfikacyjnych, obiektowych języków programowania, obiektowych baz danych, baz wiedzy oraz narzędzi CASE wyposażonych w odpowiednie repozytorium do utrzymywania definicji klas i obiektów.

Bankowość posiada pewną specyfikę, która powinna znaleźć w podstawach metodycznych odpowiednie odzwierciedlenie, aczkolwiek trudno jest dla niej znaleźć bezpośrednio odpowiedniki w terminologii obiektowego podejścia, często przykrojonego do potrzeb poszczególnych języków programowania.

2. Cechy specyficzne w sferze rozwiązania informatycznego dla bankowości

Złożoność i specyfika systemu bankowego wynika z jego następujących cech:

- skomplikowane relacje algorytmiczne pomiędzy obiektami
- duży wolumen danych głównych (przechowywanych w bazach operacyjnych i hurtowniach danych) sięgający często wielu gigabajtów
- duże znaczenie transakcji realizowanych w czasie rzeczywistym
- duży wolumen i intensywność napływu transakcji (szczególnie w bankowości detalicznej),
- samodzielność transakcji: wiele transakcji jest utrzymywanych tak jak rekordy główne baz danych, np. transakcje dilerskie, kolejne lokaty itp. rozliczane są w czasie, przechowywane w bazach operacyjnych do momentu ustania ich aktywności,
- występowanie transakcji z datami efektywnymi (do przodu lub do tyłu)
- różnorodność typów transakcji: (dla banku uniwersalnego ponad 1000)
- globalny charakter transakcji (krajowy i międzynarodowy) a więc kontrahenci (banki) i klienci banków pochodzić mogą z dowolnego kraju, w banku stosowane są różnorodne waluty, wahania ich kursów oraz sytuacja klientów wpływają na pozycję finansową banku
- występowanie *pakietów zewnętrznych* (np. obsługa SWIFTa, specjalizowane pakiety
- pochodzące od innych wytwórców ..)
- *duża złożoność klas*, np. *wiele algorytmów utrzymywania produktów tej samej klasy* np. każda lokata w ramach klasy depozytów może mieć inne szczegóły rozliczeniowe (stopy procentowe, okresy, zasady kapitalizacji itp.) przy równoczesnej możliwości stosowania polimorficznych mechanizmów w stosunku do wielu klas (np. obliczanie odsetek).

¹Materiały konferencyjne "Zastosowania rozwiązań informatycznych w bankowości"

Wyd. PN766 Akademii Ekonomicznej we Wrocławiu 1997 r. s.193-206

² jak wiadomo, polimorfizm polega na tym, że pod tą samą nazwą kryją się różne „rzeczy” w zależności od typu obiektu, który ją odziedziczy

- *wysokie obciążenie* zarówno obsługą transakcji w czasie rzeczywistym jak i wsadowym przetwarzaniem (podczas tzw. zamykania dnia oraz dla potrzeb zarządzania finansami banku). Narzuca to konieczność podziału zasobów informacyjnych na bazy operacyjne i hurtownie danych.
- realizacja transakcji własnych oraz w imieniu klientów.
- wysoki stopień ryzyka finansowego wymaga jego pomiaru (w tym prognozowania) na populacji wielu obiektów zarówno za pomocą metod statystycznych jak i nieliniowych (opartych np. na modelach neuronowych).
- wymaganie *bezbłędności* działania oprogramowania aplikacyjnego i zabezpieczenia przed nieuprawnionym dostępem (tam gdzie są pieniądze wszystko jest możliwe).

Zapewnienie bezbłędного działania wielu (np. tysiąca) programów w krytycznych warunkach przetwarzania (czas rzeczywisty, tysiące terminali rozproszonych w sieci, ryzyko finansowe prawie każdej transakcji) stanowią wyzwanie nieporównywalne do tradycyjnych aplikacji typu systemy płacowe i gospodarka materiałowa.

3. Obiektowe podejście

W warunkach obiektowego podejścia właściwą metodą wykrywania i definiowania obiektów oraz dotarcia do istoty ich zachowania się powinno być modelowanie działalności biznesowej. Oznacza to jakościową zmianę etapu przygotowawczego zwanego zwykle „analizą dotychczasowego systemu” i ograniczającego się do analizy przepływu danych, opisu danych i zebrania algorytmów. Oczywiście, dyskusyjną jest sprawą na ile modelowanie biznesu jest podparte dobrymi narzędziami, ale ostatecznie (przy minimalnych wymaganiach) nie chodzi tutaj o symulację procesów decyzyjnych (wraz z obliczaniem ich skutków), lecz o dokładne zrozumienie specyfiki (business mission) działalności jaką prowadzi przedsiębiorstwo i możliwość uzyskania takiego obrazu procesów biznesowych, aby stało się możliwe ich obiektowe sklasyfikowanie i uchwycenie mechanizmów wielokrotnego (reusable) użytku.

W przypadku działalności bankowej poza uchwyceniem istniejących produktów bankowych warto poznać ich tendencje rozwojowe poprzez odwołanie się do strategii finansowej banku w zakresie pozyskiwania i inwestowania kapitału. Strategia ta również wymaga odpowiedniego wsparcia informatycznego, np. poprzez obliczanie przepływu pieniężnego do przodu oraz ryzyka bankowego.

Wymagane jest więc szerokie spojrzenie aby w miarę kompletnie zdefiniować występujące w systemie klasy oraz mechanizmy łączności pomiędzy nimi, aby móc następnie budować bibliotekę wspólnych elementów i sukcesywnie tworzyć oprogramowanie obsługi obiektów, zachowując konsekwencję dziedziczenia cech (w tym operacji) przez dodatkowo wprowadzane podklasy i wszystkie obiekty znajdujące się wewnątrz nich. W ten sposób - w odróżnieniu od klasycznego projektowania - wprowadzenie dodatkowego obiektu „mieszczącego się w jakiejś klasie” nie wymaga jego zdefiniowania i oprogramowania od początku lecz - po przejściu wszystkich elementów „wspólnych” klasy i podklasy - wprowadzenia jedynie mechanizmów specyficznych. Podstawą prawidłowego zdefiniowania obiektów powinien być model funkcjonowania instytucji, co jest zadaniem złożonym i pracochłonnym zważywszy na skomplikowany charakter działalności bankowej.

Oczywiście zdefiniowanie i zbudowanie repozytorium (magazynu) „wspólnych typowych elementów” w przypadku systemu bankowego nie jest łatwe. Produktów i Instrumentów finansowych w banku uniwersalnym jest wiele (zwykle kilkaset, wzięwszy pod uwagę również papiery wartościowe oraz instrumenty pochodne) i są one ze sobą powiązane na poziomie zarządzania skarbowością (obracają tym samym surowcem jakim jest pieniądź). Liczba samodzielnych funkcji wyrażanych poprzez czynniki menu sięgać może ponad 5000, nie licząc wielu czynności wykonywanych automatycznie przy obsłudze transakcji (np. generowanie księgowania) oraz przy zamykaniu dnia. Ponadto nie należy zapominać o tym, że elementy repozytorium mają naturę programistyczną, a więc wymagają starannego testowania, gdyż będą one przejmowane przez wiele obiektów. Nadmienimy jeszcze dla porządku, iż niezbędne jest użycie narzędzi oraz obiektowych języków programowania (np. Eiffel, Smalltalk) do tworzenia obiektowych aplikacji. Im szerszy jest kontekst definiowania klasy (a więc im szerszy zakres aplikacyjny) tym większy będzie stopień późniejszego wielokrotnego (reuse) wykorzystania mechanizmów zdefiniowanych w klasie, ale i wyższy koszt prac analitycznych.

Generalnie rzecz biorąc, w każdym systemie można wyróżnić pewne obiekty charakterystyczne. Jako kryterium wyodrębnienia obiektu przyjmuje się *jego specyficzne zachowanie* (czyli model działania), a nie tylko to, że posiada określony zestaw atrybutów (danych) oraz to, że

jest jednoznacznie identyfikowalny.. Jednym z elementów zachowania się obiektu są jego relacje z innymi obiektami, a w szczególności typ relacji (np. jedno- lub dwukierunkowe) i jej krotność zwana cardinality lub multiplicity (1:N, N:N, 1:0, 1:1 ...), warunkowość lub bezwarunkowość, przechodność, symetryczność/asymetryczność, relacje całości do części itp.), a przede wszystkim zdarzenia, w których uczestniczy (lub które generuje) obiekt. Podstawą zdefiniowania obiektów powinien być więc przede wszystkim model funkcjonowania a nie tylko statyczne struktury danych.

Nie podejmujemy się ścisłego zdefiniowania obiektu. W szerokiej definicji obiekt jest scaleniem (enkapsulacją, hermetyzacją) danych, procedur i reguł zachowania się wynikających z definicji klasy, stanowi więc jakby przeciwieństwo filozofii tradycyjnych systemów zarządzania bazami danych, w których króluje tzw. niezależność danych (data independence) czyli rozdział definicji danych od programów aplikacyjnych._

Klasa właściwie jest definicją właściwości grupy obiektów i służy do ich generowania. *W działalności bankowej istnieje co najmniej kilkanaście (może nawet kilkadziesiąt) takich grup.* Przykładowo, klasą może być grupa rachunków depozytowych, grupa rachunków kredytowych, grupa płatności zagranicznych dokumentowych (akredytyw) itp. W zależności od potrzeb (czyli złożoności) klasy możemy dzielić na podklasy, te na grypy itp. Przykładowo, w klasie kredytowej wyróżnić można podklasę rachunków kredytowych, podklasę sposobu spłaty kredytu oraz podklasę opłat i kar. Obiekt dziedziczy charakterystykę po klasie i posiadać może dodatkowo swoje własne cechy. Obiekt jest końcowym elementem dekompozycji klasy, stanowi więc referencje (odzworowanie) realnie istniejących obiektów rzeczywistych (obojętnie czy są one fizyczne czy np. jako zapisy) w tym sensie, że zawiera dokładny opis ich struktury i zachowania.

W naszej konwencji **obiekt jest wystąpieniem klasy na ostatnim poziomie** jej podziału strukturalnego (superklasa, klasa, podklasa, obiekt), czyli pozostaje nadal pojęciem abstrakcyjnym (specyfikacyjnym), służąc do wyznaczenia typu (grupy wystąpień obiektów rzeczywistych z tą samą charakterystyką typowości) dziedziczoną zgodnie z drzewem genealogicznym (od superklasy). Dopiero wystąpienie obiektu jest fizycznym „egzemplarzem” i odnosi się do indywidualnej jednostki (np. rachunek bieżący Pana X. Jeśli obiektem jest książka o danym tytule, to każdy egzemplarz tej książki jest wystąpieniem obiektu. Wystąpienie obiektu identyfikowane jest w tym przypadku przez tytuł i jego kolejny nr. Pełny ciąg dekompozycyjny wygląda więc następująco: superklasa, klasa, podklasa, obiekt, fizyczne wystąpienie obiektu.

Gdybyśmy przyjęli konwencję, że pojęcie obiektu jest bezpośrednio reprezentacją fizycznego egzemplarza, wówczas zabrakłoby pojęcia dla „elementu” abstrakcyjnego jednoznacznie określonego przez wszystkie definicje dziedziczone poprzez dekompozycję klasy. *Być może, gdyby przyjąć w miejsce „obektu” termin „encja” (entity), wówczas „obiekt” mógłby zejść do poziomu fizycznej reprezentacji każdego wystąpienia.* Nie zrobiliśmy tego z tej przyczyny, że encja często utożsamiana jest z klasą, ale w zasadzie nie mamy nic przeciwko temu. Wystąpienie obiektu można nazwać ew. fizycznym obiektem, ale to z kolei może może zaburzyć klasyfikację obiektów w modelowaniu instytucji (obiektami fizycznymi są klienci, produkowane wyroby, itp. a abstrakcyjnymi np. produkty bankowe, przepływy pieniężne, brokerzy (softwareowi) w architekturze klient/serwer i agenci (software’owi) reprezentujący obiekt

Klasy (a więc i obiekty) definiuje się poprzez ustalenie nazwy (identyfikatora klasy i zasad identyfikacji obiektów tworzonych w ramach klasy), metod (operacji, funkcji, procedur) jej przynależnych, atrybutów (listy danych), reguł (zasobów, parametrów) tworzenia obiektów, mechanizmów zewnętrznej komunikacji (eksportowo-importowych), komunikatów, wyjątków, zdarzeń (np. transakcji, kontraktów) zachodzących na obiektach klasy itp.

Po zdefiniowaniu klasy każdy obiekt będący jej wystąpieniem działa na zasadach czarnej skrzynki (zgodnie z zasadą enkapsulacji) - nie musimy znać jego wewnętrznych mechanizmów ale znamy dokładnie jego zachowanie (reakcje na dochodzące do niego impulsy). Klasycznym przykładem obiektu jest samochód, o którego konstrukcji możemy prawie nic nie wiedzieć, ale wiemy jak zareaguje na naciśnięcie gazu czy skręcenie kierownicą i możemy koncentrować uwagę wyłącznie na tym jak pokonać trasę przejazdu bez kolizji z innym pojazdem.

Podobnie w dobrze obiektowo zdefiniowanym systemie bankowym, znając zachowania się produktów bankowych możemy koncentrować się na przykład na problemach ryzyka bankowego, a więc sprawach ich współdziałania poprzez rozwarcie stóp procentowych, badanie przepływu pieniężnego, itp. W zależności od wyników analizy możemy tworzyć nowe obiekty (nowe produkty

bankowe), wysyłając do klasy nowy zestaw parametrów (np. okres lokaty/kredytu, stopy procentowe itp.), czyli wykonywać to co nazywamy definiowaniem i modyfikacją produktów bankowych.

Objektowa orientacja analizy i projektowania wymaga więc działań na klasach i obiektach, zakładając mechanizmy dziedziczenia i ich wzajemnego komunikowania się.

Przenosząc obiektowe zasady do sfery projektowania systemów (czyli w uproszczeniu mówiąc - zamieniając obiekty biznesowe na obiekty software'owe) możliwa staje się budowa systemu bankowego oparta na wielokrotnym montażu w różnych zestawach software'owych obiektów tej samej klasy wyposażonych w odpowiednie łącza. Obiekty występują w systemie tylko jednokrotnie (ewentualnie w postaci automatycznie tworzonych kopii), zaś każdorazowe ich użycie odbywa się wg tych samych mechanizmów, zapewniając nieredundancję i spójność. Przykładem obiektu wielokrotnego użycia może być procedura obliczania odsetek, używana „polimorficznie” (z tą samą nazwą ale z różnymi danymi i metodami) w kilku modułach systemu (rachunkach bieżących, depozytowych i kredytowych).

W zaawansowanych rozwiązaniach obiektowych obiekt stanowić może enkapsulację (wbudowanie) takich charakterystyk jak formy danych (np. multimedialne), struktury i relacje danych, reguły wnioskowania (jako wspomaganie baz wiedzy w systemach eksperckich), zaawansowane słowniki i reguły gramatyczne (wspomagające przetwarzanie języków naturalnych oraz rozpoznawanie mowy), reguły zachowania się obiektu w procesach i zdarzeniach zachodzących w rzeczywistości.

Nowoczesny obiektowy system bankowy oparty powinien być więc na semantyce umożliwiającej symulację działalności finansowej banku. Wymagania tego nie spełniają komercyjnie oferowane systemy zarządzania bazami danych (w tym relacyjne bazy danych).

Obiektom stawiane są następujące podstawowe wymagania formalne:

- dobrze określone granice (nie mogą zachodzić na obiekty należące do innych klas)
- określony typ zachowania w scenariuszu OLH (Object Life History) czyli znane są zdarzenia związane z obiektem (generowane przez niego lub dochodzące do niego)
- typowość czyli przynależność do klasy (dziedziczenie cech klasy)
- nie może być to byc chwilowy element systemu o nieokreślonym zachowaniu
- unikalna identyfikowalność (najlepiej gdyby była niezależna od zmian wartości pól kluczowych i atrybutowych w bazach danych - to czy obiekt istnieje np. departament nie zależy od tego, czy zmieni mu się symbol komórki organizacyjnej)

Z obiektami stowarzyszona jest ścieżka wywołań procedur, odwołania do tablic (np. księgowani), odwołania do transakcji, formatki ekranowe, struktury danych, odwołania do notatek i korespondencji towarzyszącej (np. do przypomniaczy zwanych ticklerami, wyciągów, wniosków kredytowych) itp.

Rozróżnić można obiekty elementarne, złożone, dynamiczne i informacyjne.

W grupie obiektów elementarnych występują:

1. podmioty (klient, bank, oddział banku)
2. rachunki (klientów, papierów wartościowych)
3. kontrakty na otwarcie rachunków, na dokonanie transakcji dilerkiej itp.
4. konta księgowo
5. waluty i rachunki własne NOSTRO
6. limity (może być też traktowany jako obiekt informacyjny)
7. centra kosztów/dochodów np. oddziały, departamenty centrali itp.
8. użytkownicy systemu.

Obiekty elementarne są zarówno obiektami jak i atrybutami. Limit musi być wyrażony w walutach i dotyczy określonych klientów czy banków albo kontraktów. Limit staje się obiektem gdy podlega kontroli jako „samodzielny” element: np. globalny limit kredytowy klienta, globalny limit waluty itp. Podobnie waluta jest obiektem np. w rachunku strat i zysków waluty. Występowanie tej samej nazwy obiektu w różnych kontekstach (jako atrybuty) służy do wskazywania powiązań integracyjnych i

agregacyjnych (np.dla hurtowni danych). Byłoby zupełnie źle gdyby te same obiekty miały być identyfikowane przez różne nazwy w różnych aplikacjach !, co nie oznacza, że dla wygody użytkowników nie mogą oni używać swoich własnych synonimów (np. w swoich językach narodowych) na poziomie użytkownika systemu (w rdzeniu systemu czyli centralnym repozytorium istnieć muszą identyfikatory pierwotne).

W grupie obiektów złożonych występują przede wszystkim produkty bankowe np. depozyty, kredyty, akredytywy, papiery wartościowe itp. Z natury swojej obiekty złożone odwołują się do obiektów elementarnych (rachunków, klientów, transakcji), stanowiąc wyższy poziom hierarchiczny.

W grupie obiektów dynamicznych występują :

1. *zdarzenia*

Zdarzeniem jest elementarny niepodzielny fakt, czyn, działanie.

Zdarzenia pochodzą z zewnątrz lub powstają wewnątrz systemu i zwykle powodują zmianę stanu obiektu np.niedokonanie spłaty raty kredytu w terminie zmienia status kredytu.

2. *transakcje*

Transakcją jest sekwencja zdarzeń mająca na celu realizację krótkoterminowego celu, np. uzgodnienie warunków kontraktu lokaty i dokonanie wpłaty na lokatę, przelewy, transakcja forward, spot, swap itp.

3. *akcje*

Akcją jest zwykle złożone działanie dotyczące obsługi transakcji lub obsługi żądania informacji wynikowej o złożonym algorytmie lub wymagającej przetwarzania wielu zapisów w bazie danych lub wielu transakcji, np. deal confirmation matching, nostro account reconciliation, ocena kategorii kredytowej i tworzenie rezerw na trudne kredyty.

4. *procesy*

Procesami są złożone operacje dotyczące obiektów w całym obszarze funkcjonalnym np. revaluacja transakcji zagranicznych.

Poza wyżej wymienionymi grupami można definiować w systemie bankowym klasy obiektów nie będące bezpośrednim odwzorowaniem działalności bankowej. Należą do nich obiekty informacyjne.

W grupie obiektów informacyjnych wyróżnia się zestawy informacji posiadające podstawowe znaczenie dla decyzji bankowych lub konieczne ze względu na bezpieczeństwo systemu:

:

- 1.pozycja klienta
- 2.pozycja waluty
- 3.pozycja banku
- 4.przepływ pieniężny (cash flow)
- 5.bilans księgowy
- 6.ślady audytowe (audit-trial journal).
- 7.limity

Są to albo węzły informacyjne, zasilane przez większość (jeśli nie wszystkie) transakcji albo przekroje informacyjne wymagające odpowiedniego stanu zintegrowania informacji, rzutuące na rozwiązania architektury systemu.

Procesy w systemie bankowym mogą być inicjowane manualnie lub automatycznie z wielu przyczyn (w zależności od typu transakcji, produktu, typu klienta, upływu czasu itp.).

Procesy w systemie bankowym mogą być inicjowane z wielu przyczyn (transakcja, produkt, procedura bankowa, typ klienta, moment czasowy).

Transakcje mogą być zewnętrzne (np.wpłata gotówki na rachunek) lub też wewnętrzne tj. generowane przez system zgodnie z założoną dynamiką systemu (np.naliczanie odsetek zgodnie z datą zapadalności, uruchomienie transakcji dilerskiej forwardowej w wyznaczonym dniu itp.) Czynnik czasu stanowi jeden z najistotniejszych cech systemów bankowych i jego ilustracją najlepszą jest raportowanie tzw. cash flow.

Produktom bankowym podporządkowane są transakcje, których sekwencja tworzy procesy. Dookoła produktu występuje więc specyficzne środowisko, obejmujące sekwencje transakcji oraz księgowości.

W projektowaniu obiektowym klasa jest formą nadrzędną służącą do produkcji obiektów. Obiekt jest więc wystąpieniem elementu klasy, dziedziczącym jej właściwości, posiadającym dobrze określone granice fizyczne i funkcjonalne (wartości atrybutów i funkcje), dzięki czemu możliwe jest formalne zdefiniowanie mechanizmów tworzenia i opuszczania (niszczenia) obiektu (w programowaniu obiektowym zwanych konstruktorami i destruktorami). Znaczenie obiektów dla metodyki projektowania i technologii informatycznych jest radykalne, jeśli spojrzeć na to od strony integracji systemu. Obiektami są bowiem elementy systemu *widoczne z każdego poziomu architektonicznego* i bezpośrednio wykorzystywane w wielu miejscach bez potrzeby dublowania danych i funkcji oraz tworzenia sztucznych mechanizmów integracyjnych pomiędzy modułami systemu (np. w postaci konwersji rekordów na pliki tekstowe i odwrotnie). Interfejsy stanowią bowiem integralną część obiektu. Przykładem wykorzystania obiektu w programowaniu wizualnym (visual object programming) jest najechanie myszą na ikonę obiektu i „wciągnięcie” go do pisanego modułu/programu, zamiast pisania kodu programowego.

Transakcje bankowe są podporządkowane zwykle produktom bankowym, tzn. mogą występować tylko w określonych procesach. Np. założenie lokaty związane jest z produktem lokaty, itp. Szereg transakcji (np. księgowania na kontach syntetycznych) występuje jednak w wielu produktach.

W każdym procesie występują obiekty elementarne albo jako atrybuty albo jako podmioty. Obsługa tych obiektów, a ściślej rzecz biorąc sposób ustalania relacji pomiędzy nimi w przekroju całego systemu bankowego, decyduje o jakości architektury systemu.

Możliwość wyznaczania relacji w sposób elastyczny oraz zapewnienia zmienności obiektów (poprzez wprowadzanie parametrów) jest prawdopodobnie najtańszą i najszybszą metodą reorganizacji systemu bankowego i jego adaptacji do potrzeb rynku, nieporównywalnie lepszą od klasycznych zmian organizacyjnych w długim cyklu decyzyjnym (zlecenie oprogramowania nowych produktów, powoływanie nowych komórek organizacyjnych do obsługi nowych produktów, publikowanie regulaminów nie tyle dla klientów ile do szkolenia własnego personelu itp.) Skrócenie procesu decyzyjnego (przy równoczesnym lepszym zabezpieczeniu informacyjnym) oznacza również skrócenie drogi pomiędzy klientem i bankiem.

Relacje na poziomie kont księgowych wyznaczane są stosunkowo prosto poprzez schematy księgowości oraz powiązania miejsc powstawania kosztów/dochodów. Do uzyskania innych relacji zalecane jest użycie obiektowych baz danych, a przynajmniej np. relacyjnego albo hierarchicznego systemu zarządzania bazą danych, a czasem nawet zaawansowanej technologii data-mining opartej na regułach odkrywania wiedzy.

Wymienione wyżej *obiekty jak nić przewodnia oplatają system bankowy, przechodząc przez wiele modułów aplikacyjnych*. Nienależyte uwzględnienie któregośkolwiek z tych elementów mocno obniża jakość systemu. Na przykład w niewielu systemach występuje możliwość definiowania transakcji i stowarzyszonych z nimi księgowości. W innych nie ma obsługi limitów, albo zawężone one są tylko do poszczególnych produktów.

Limity dotyczą kraju, regionu, branży, klienta (limit globalny - management limit - dla klienta i ew. jego macierzystej korporacji, limity przyznane - granted limits - dla kredytów, akredytyw, ujemnego salda na rachunku, itp.), waluty, typów transakcji (forex, rynek pieniężny, papiery wartościowe...). Przy okazji uwaga, iż suma limitów przyznaczonych może być większa od limitu globalnego, zaś limitów wykorzystanych powinna być mniejsza od niego. List limitów w systemie powinno być wiele, aby można było spełnić różnorodne wymagania banków. Limity powiązane są z zabezpieczeniami w tym blokowaniem depozytów, papierów wartościowych itp.

Obsługa limitów polega m.i. na:

- sygnalizacji przekroczeń limitów podczas wprowadzania transakcji,
- bieżącej aktualizacji ich sald (w układzie limitów ustalonych, wykorzystanych i dostępnych), raportowaniu przekroczeń i wykorzystania limitów (na jakie cele

zostały użyte),

- dostarczaniu analitycznych informacji do analizy przekroczeń limitów (wykaz wszystkich dokonanych transakcji wg podmiotu limitu, oddziałów, dilerów),
 - raportowaniu limitów które tracą ważność itp.
- Raporty (lub odpowiedzi na zapytania) powinny być uporządkowane wg bankowców (opiekunów), klientów itp.

Waluta i klient oraz czas stanowią nieodłączny atrybut prawie każdego zdarzenia w systemie. Obliczanie pozycji klientów i walut wymaga ustalenia w architekturze systemu "ścieżek" dotarcia do wszystkich transakcji finansowych (czyli do wnętrza modułów aplikacyjnych).

Wprowadzenie zmian projektowych dotyczących tych obiektów jest wobec tego bardzo pracochłonne i odpowiedzialne. Dlatego kupując system trzeba ocenić na ile zaawansowana jest architektura systemu w zakresie orientacji na klienta, wielowalutowość, wielorakość limitów, definiowalność transakcji itp.

4. Parametryzacja obiektów

System bankowy powinien być modyfikowalny nie tylko podczas instalacji, lecz również podczas użytkowania m.i. za pomocą narzędzi służących do definiowania i redefiniowania produktów bankowych.

Możliwość zapewnienia zmienności obiektów (np. poprzez ich definiowanie za pomocą parametrów) jest prawdopodobnie najtańszą i najszybszą metodą reorganizacji systemu bankowego i jego adaptacji do potrzeb rynku, nieporównywalnie lepszą od klasycznych zmian polegających na dokonywaniu zmian w kodach źródłowych programów.

Metody parametryzacji bywają różne. Z reguły sprowadzają się one do nadania wartości parametrom w trakcie wprowadzania kontraktu (poprzez akceptację wartości domyślnej lub dziedziczonej z definicji klasy produktów, albo wprowadzenie nowej wartości) wchodzącym w zakres definicji danej klasy produktów. Parametrów takich dla każdego typu produktu może być nawet kilkaset. Poprzez parametryzację można definiować również zakres (np. rodzaje transakcji) limitów i pozycji (klienta, waluty i banku).

Parametry zwykle są przechowywane w postaci wielu tablic (plików). Korzystne byłoby - z punktu widzenia administrowania systemem - aby odwołania do nich odbywały się poprzez specjalną bankową metabazę danych, kontrolującą wzajemną spójność tablic i produktów

Im głębsza parametryzacja systemu tym bardziej system powinien być wyposażony w mechanizmy automatycznego wykrywania ewentualnych wewnętrznych sprzeczności, utrzymywania niezbędnych relacji pomiędzy tablicami itp. Bez takiego podparcia trudno jest system przetestować w pełni na etapie wdrożenia i gwarantować jego bezbłądność podczas przetwarzania (gdy np. pojawią się nietestowane kombinacje parametrów lub nietestowane funkcje). Pożądane jest dla stabilności systemu, aby na przykład w ramach procedur otwierania dnia uruchamiany był specjalny przebieg testujący spójność systemu (aby nieoczekiwane „run-time errors” nie pojawiały się w trakcie obsługi klientów).

Obiektem w systemie bankowym jest również jego użytkownik. W odpowiednio elastycznych rozwiązaniach istnieje możliwość prawie dowolnego konfigurowania funkcji dla poszczególnych użytkowników poprzez generowanie menu w ramach klas użytkowników (włącznie z hierarchicznym zagnieżdżaniem oraz ustalaniem zabezpieczenia hasłowego). Konfiguracja funkcjonalna systemu może być wówczas dokładnie dopasowywana do zakresu działania stanowisk pracy.

Ustalanie reguł funkcjonowania np. debetowych i kredytowych rachunków bankowych następuje poprzez tzw. definiowanie produktów bankowych oraz transakcji. Różnicowanie produktów zależy nie tylko od waluty, lecz wielu innych elementów zwanych parametrami. Zaletą parametryzacji jest przede wszystkim to, że nie wymaga ona zmian w kodzie źródłowym oprogramowania, a więc znajduje się całkowicie w gestii banku.

Definiowanie produktów wygląda rozmaicie w poszczególnych systemach . Ujmując to zagadnienie najszerzej można powiedzieć , że definicje te powinny przebiegać na poziomie klas i typów produktów, uwzględniając parametry globalne na poziomie banku jako całości („dziedziczone” domyślnie jako parametry klasy produktów). Niektóre systemy dopuszczają różnicowanie typów produktów dla regionów i oddziałów oraz dają możliwość definiowania transakcji bankowych, zdarza się to niestety stosunkowo rzadko.

Środowisko elastycznego systemu może być bardzo szerokie obejmując możliwości aplikacji, narzędzi i systemu zarządzania bazą danych.

PARAMETRYZACJA OBIEKTU: RACHUNEK BANKOWY

(głównie dla depozytów i kredytów)

- waluta rachunku,
- typy stop oprocentowania (stałe, zmienne),
- podstawa obliczeń (np. $actual/360, 30/360, 360/360, 30/365$ i 366),
- nazwa indeksu (tabeli) stóp procentowych
- odwołania do tablicy księgowañ lub księgowania bezpośrednio wpisywane do definicji,
- minimalne saldo do obliczania odsetek
- rozpiętość stóp (rate floor, rate ceiling)
- możliwość konwersji typu stóp (np. ze zmiennej stopy na stałą po upływie 2 lat)
- warunki odnowienia lokaty
- limit dla rachunków kredytowych
- wskaźnik („flaga”) dopuszczający saldo negatywne (overdraft) na rachunku typu ROR
- połączenia z innymi rachunkami (wzajemna kompensacja sald negatywnych, kredytów, opłat itp.),
- moment startu oprocentowania
(w lokatach terminowych - od pierwszego lub następnego dnia, w innych produktach- wg daty transakcji, wg daty walutowania, wg daty księgowania itp.)
- tryb spłaty kredytu (z góry lub z dołu miesiąca, itp.)
- minimalna i maksymalna kwota odsetek,
- algorytm spłat kapitału i odsetek (dla kredytów)
- tryb spłat zaległych płatności (dla kredytów)
- warunki spisania na straty ,
- okres odsetkowy i okres kapitalizacji (np. 1 dzień i 1 rok dla rachunków a ' vista),
- podstawa obliczania odsetek (saldo przed transakcją, saldo po transakcji, kwota transakcji, limit),
- typ opłat i prowizji (kwotowe zależne od przedziałów, procentowe itp.)
- możliwość zmiany stopy procentowej w trakcie trwania umowy,
- możliwość wypłaty odsetek bez zrywania umowy ,
- możliwość automatycznego odnowienia (automatic rollover),
- zabezpieczenie kredytu (kwotowe lub procentowe),,
- sprzedaż lub kupno kredytu (participated or purchased loans),
- dopuszczalne typy transakcji stowarzyszonych z danym typem produktu,
- księgowania (lub odwołania do tablicy księgowañ, stowarzyszonych z typem produktu i typem transakcji),
- język narodowy używany do wydruku stowarzyszonych dokumentów (wyciągi, potwierdzenia operacji),
- częstotliwość wydruku wyciągu (po dniu w którym wystąpiły operacje, na koniec miesiąca - w ostatnim dniu roboczym lub kalendarzowym, raz na dwa tygodnie),
- ilość dni przechowywania historii rachunku ,
(ew. odrębne parametry dla rachunków aktywnych i rachunków zamkniętych)
- data startu typu produktu ,
- data upływu terminu ważności typu produktu,
- itd.

PARAMETRYZACJA OBIEKTU: BANK UŻYTKUJĄCY SYSTEM

(tzw. parametry globalne , obowiązujące w całym systemie)

- identyfikator banku

- symbol waluty bazowej,
- zakres tolerancji kursów walutowych (służy do kontroli prawidłowości wprowadzanych kwot dwóch walut),
- struktura organizacyjna banku (centrala, regiony, oddziały krajowe i zagraniczne, filie, punkty agencyjne, ATM),
- ilość dni do uśpienia rachunku,
- ilość dni przechowywania historii stóp oprocentowania (wyznacza to możliwą ilość dni cofania się do tyłu)
- ilość dni archiwowania księgowania,
- dopuszczalność tzw. masowych zmian do tyłu na rachunkach klientów,
- data do której można robić księgowania dotyczące zeszłego roku,
- algorytm cyfry kontrolnej,
- stopnie konsolidacji księgi głównej np. na poziomie oddziałów i centrali,
- długość roku finansowego - nie musi pokrywać się z rokiem kalendarzowym), itp.
- tabele transakcji aktualizujących pozycję banku.

PARAMETRYZACJA OBIEKTU : KLIENT

- identyfikator klienta
- waluta bazowa klienta
- typ: osoba fizyczna, prawna
- kategoria: zwykły, uprzywilejowany (np. podlegający pod private banking)
- grupa zdolności kredytowej
- typy wyciągów
- wykaz transakcji aktualizujących pozycję klienta

PARAMETRYZACJA OBIEKTU : WALUTA

- symbole (AAA,999)
- kraj
- zakres waluty: bazowa banku, lokalna oddziału
- liczba miejsc po przecinku (dla kwot)
- liczba miejsc po przecinku dla kursu wymiany
- liczba dni obsługi transakcji
- stosowalność waluty (operacje gotówkowe, polecenia przelewu krajowe i zagraniczne ,itp.)
- cechy wymiennosci
- najniższa jednostka nominalu
- nominały (banknoty i bilon)

PARAMETRYZACJA OBIEKTU : LIMIT

podmiot limitu: diler, kontrakt/produkt/transakcja, waluta, klient, branża, kraj

PARAMETRYZACJA OBIEKTU :KONTO KSIĘGOWE

- identyfikator konta (i jego struktura)
- typ konta (bilansowe po stronie aktywów/po stronie pasywów, pozabilansowe jednostronne, pozabilansowe dwustronne, konto rozliczeniowe dochodów/kosztów, konto nagłówkowe, itp.)
- waluta

Zamiast wniosków

Definicje produktów trzymane powinny być w odrębnej bazie danych, znajdującej się pod specjalnym nadzorem. Modyfikacje produktów wynikające ze strategii finansowej banku wymagają określonej autoryzacji w ramach organizacyjnych procedur zarządzania zmianami (kontrola uprawnień, dokumentowanie zmian i testowanie). Baza produktów przechowywać powinna szereg wersji z różnymi datami efektywnymi (lub datami dystrybucji). Wprowadzanie nowych klas produktów następuje zwykle w fazie tzw. otwierania dnia a nie w dowolnym momencie w ciągu dnia.

Szerokim możliwościom systemu do definiowania nowych typów produktów musi towarzyszyć też odpowiednia kontrola ryzyka finansowego produktów (dotyczy to szczególnie kredytów, papierów wartościowych oraz instrumentów derywatywowych).