

# Podejście obiektowe

## w modelowaniu procesów gospodarczych i systemów informacyjnych przedsiębiorstw

**Mirosława Lasek**, prof. UW

Katedra Informatyki Gospodarczej i Analiz Ekonomicznych WNE UW

**Marek Pęczkowski**, mgr

Katedra Informatyki Gospodarczej i Analiz Ekonomicznych WNE UW

### Wstęp

Podejście obiektowe powstało i rozwinęło się jako podejście metodologiczne tworzenia systemów oprogramowania — podobnie jak strukturalne i społeczne — w ramach inżynierii oprogramowania [Avison, Fitzgerald, 2003; Chmielarz, 2000; Sommerville, 2001; Wrycza, 1999]. Celem jego opracowania było tworzenie systemów informatycznych, tak aby poradzić sobie ze złożonością wytwarzanego oprogramowania i procesów jego produkcji, ograniczyć koszty, spowodować wzrost wydajności wytwarzania i doprowadzić do sytuacji, w której programy spełniają wymogi użytkowników, są niezawodne, efektywne, łatwo poddające się pielęgnacji i ergonomiczne [Bluemke, 2001; Chabik, 1999; Subieta, 1999 (b)].

Podejście obiektowe jest już dobrze znane i doceniane w środowisku inżynierów oprogramowania, projektantów i programistów. Mniej uwagi poświęca się możliwościom i korzyściom, jakie może dać zastosowanie koncepcji obiektowości w modelowaniu procesów gospodarczych.

Celem artykułu jest przedstawienie podejścia obiektowego nie jako podejścia służącego potrzebom tworzenia oprogramowania, tj. implementacji projektu systemu informacyjnego w wybranym języku programowania, ale jako podejścia przydatnego w modelowaniu procesów gospodarczych i tworzeniu architektur systemów informacyjnych przedsiębiorstw, tak aby zapewnić ich optymalizację w zakresie kosztów, czasu i wykorzystania zasobów.

Na Wydziale Nauk Ekonomicznych UW dysponujemy trzema programami: *Rational Rose Enterprise Edition* firmy IBM (do niedawna firmy Rational), *ARIS* firmy IDS Prof. Scheer oraz *iGrafx Process* firmy Corel (do niedawna firmy Micrografx), które możemy wykorzystać dla zastosowania podejścia obiektowego w modelowaniu procesów gospodarczych.

## 1. Podstawowe pojęcia obiektowości w modelowaniu procesów gospodarczych, tworzeniu architektur i wdrażaniu systemów informacyjnych

W modelowaniu procesów gospodarczych mogą być wykorzystane podstawowe pojęcia obiektowości, tak jak je przyjęto w inżynierii oprogramowania [Bluemke, 2001; Chabik, 1999; Lethbridge, Laganière, 2001; Shalloway, Trott, 2002; Subieta, 1999(a); Subieta, 1999(b); Wrycza, 1999].

Do tych podstawowych pojęć należy zaliczyć pojęcie obiektu i klasy obiektów. Najprościej można określić obiekt jako rzecz lub pojęcie pochodzące z dziedziny przedmiotowej lub z przestrzeni dopuszczalnych rozwiązań [Subieta 1999(a); Subieta, 1999(b); Wrycza, 1999]. Obiekty mają tożsamość (można im przypisać unikalną nazwę), stan (określony przez atrybuty obiektu) i zachowanie (operacje, które można na nich wykonywać; ich implementacje określa się mianem metod).

Pod pojęciem klasy rozumie się cechy grupy obiektów, które są dla nich niezmiennie — tzw. inwarianty. Podstawowymi inwariantami są nazwa, atrybuty, operacje.

Metody danego obiektu są realizowane po otrzymaniu komunikatu od innego obiektu, żądającego realizacji metody. Komunikat taki składa się z identyfikatora obiektu-odbiorcy komunikatu, metody, którą obiekt-odbiorca powinien wykonać i opcjonalnie parametrów, wskazujących obiektowi-odbiorcy, jak wykonać wskazaną metodę. Jeżeli ten sam komunikat — zawierający tę samą nazwę operacji — wysłany do różnych obiektów powoduje wykonanie różniących się między sobą metod, to mamy do czynienia z polimorfizmem — tj. ta sama nazwa odnosi się do różnych implementacji operacji.

Enkapsulacja (hermetyzacja) oznacza traktowanie obiektu jako kapsuły, pełniącej określoną funkcję w systemie. Kapsuła stanowi „czarną skrzynkę”, udostępniającą tylko niezbędne informacje i ukrywającą pozostałe.

Klasy obiektów mogą tworzyć hierarchię. Klasa podrzędna (podklasa) dziedziczy atrybuty i metody klasy nadrzędnej. Nie trzeba ich więc ponownie definiować dla podklasy. Hierarchie klas są tworzone poprzez operacje generalizacji i specjalizacji. Generalizacja oznacza budowanie pojęć bardziej ogólnych na podstawie pojęć bardziej szczegółowych, natomiast specjalizacja jest budowaniem pojęć bardziej szczegółowych, wywodzących się z pojęć bardziej ogólnych.

## 2. UML w modelowaniu procesów gospodarczych

UML (*Unified Modeling Language*) jest językiem umożliwiającym modelowanie z wykorzystaniem pojęć i podejścia obiektowego. Definicję UML podaje firma Rational, producent programu Rational Rose, wspomagającego projektowanie i programowanie obiektowe [<http://www.rational.com/uml>]. W tłumaczeniu przedstawionym przez Kazimierza Subietę [Subieta, 1999(b)] wygląda ona następująco:

UML jest językiem do specyfikacji, konstruowania, wizualizacji i dokumentowania wytworów związanych z systemami intensywnie wykorzystującymi oprogramowanie.

W oryginale ma wymowę może bardziej dobitną:

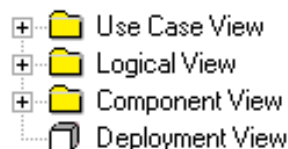
*The Unified Modeling Language (UML) is a language for specifying, constructing, visualizing, and documenting the artifacts of a software-intensive system* [<http://www.rational.com/uml>].

Z przedstawionej powyżej definicji wynika, że język UML nie odwołuje się do żadnej metodyki analizy i projektowania i nie definiuje metodyki produkcji oprogramowania. Ma charakter bardziej ogólny. Jest językiem przeznaczonym do modelowania systemów [Booch, Rumbaugh, Jacobson, 2002; Fowler, Scott, 2002; Bennett, McRobb, Farmer, 2002, Subieta, 1999(b)]. Wydaje się dobrze pasować do modelowania procesów gospodarczych i możemy zweryfikować jego przydatność do tego celu.

### 3. Programy analizy i tworzenia modeli procesów gospodarczych

#### 3.1. Program Rational Rose Enterprise Edition

Wykorzystując metodologię UML i program Rational Rose, możemy przeprowadzać analizę i modelowanie procesów gospodarczych z czterech różnych punktów widzenia (perspektyw) — jak ilustruje to rysunek 1.



#### Rys. 1.

Perspektywy modelowania dostępne w programie Rational Rose i widoczne w oknie przeglądarki programu (Browser)

Źródło: fragment ekranu z programu Rational Rose Enterprise Edition.

Możemy modelować proces, „patrzac” z punktu widzenia:

- przypadków użycia (*Use Case View*);
- logicznego (*Logical View*);
- składowych (*Component View*);
- wdrożenia (*Deployment View*).

Wszystkie wymienione punkty widzenia *Use Case View*, *Logical View*, *Component View*, *Deployment View* mogą dotyczyć tego samego procesu, ale pozwalają widzieć i modelować go z czterech różnych „stron”.

### Punkt widzenia przypadków użycia (*Use Case View*)

Rozpatrywanie z punktu widzenia przypadków użycia powinno zapewnić możliwość pełnego zrozumienia możliwości funkcjonalnych i wykorzystywania systemu. Ten punkt widzenia przedstawia wzajemne oddziaływanie na siebie aktorów (*actors*) i przypadków użycia (*use cases*).

Pod pojęciem aktora (*actor*) rozumie się:

obiekt modelujący określoną rolę zewnętrznego użytkownika systemu. Aktor może operować na innych obiektach, ale sam nie podlega operacjom ze strony innych obiektów [Subieta, 1999(a)].

Nie musi być, choć może, człowiekiem — np. użytkownikiem systemu. Może być fragmentem programu o dużym stopniu autonomii. Przypadek użycia (*use case*) jest to

pewna nazwana lub dobrze określona interakcja pomiędzy użytkownikiem a systemem komputerowym. Przypadek użycia odwzorowuje pewną funkcję systemu w taki sposób, w jaki będą ją widzieć jego przyszli użytkownicy [Subieta, 1999(a)].

W modelowaniu z punktu widzenia przypadków użycia wykorzystywane są następujące diagramy:

- przypadków użycia (*Use-case diagrams*);
- sekwencji (*Sequence diagrams*);
- współpracy — zwane też diagramami kolaboracji lub współdziałania (*Collaboration diagrams*);
- aktywności (*Activity diagrams*).

### Punkt widzenia logiki działania systemu (*Logical view*)

Ten punkt widzenia przedstawia wymagania funkcjonalne systemu. Ukazuje klasy (*classes*) i relacje między klasami. Klasa (*class*) jest definiowana jako [Subieta, 1999(a)]:

(1) zbiór obiektów o zbliżonych własnościach, (2) byt semantyczny rozumiany jako miejsce przechowywania takich cech grupy podobnych obiektów, które są dla nich niezmiennie (np. zestawu atrybutów, nazwy, metod, ograniczeń dostępu), (3) wyrażenie językowe specyfikujące budowę obiektów, dozwolone operacje na obiektach, ograniczenia dostępu, wyjątki itd.

Wykorzystywane są diagramy:

- klas (*Class diagrams*);
- stanów (*Statechart diagrams*).

#### Punkt widzenia składowych (*Component View*)

Ten punkt widzenia przedstawia organizację oprogramowania systemu. Zawiera informacje o oprogramowaniu, programach wykonywalnych i bibliotecznych składnikach systemu.

Do opisu procesów z tego punktu widzenia wykorzystywane są tylko diagramy składowych, nazywane też diagramami komponentów (*Component diagrams*).

#### Punkt widzenia wdrożenia (*Deployment View*)

Dotyczy on rozlokowania procesów do realizacji na sprzęcie komputerowym. Ten punkt widzenia wymaga szczególnej uwagi w przypadku systemów środowiska architektury rozproszonej (aplikacje i serwery w różnych lokalizacjach). Jest tu wykorzystywany jeden typ diagramu, zwany diagramem wdrożenia, czasem też nazywany diagramem wdrożeniowym lub rozprzestrzeniania.

### 3.2. Program ARIS

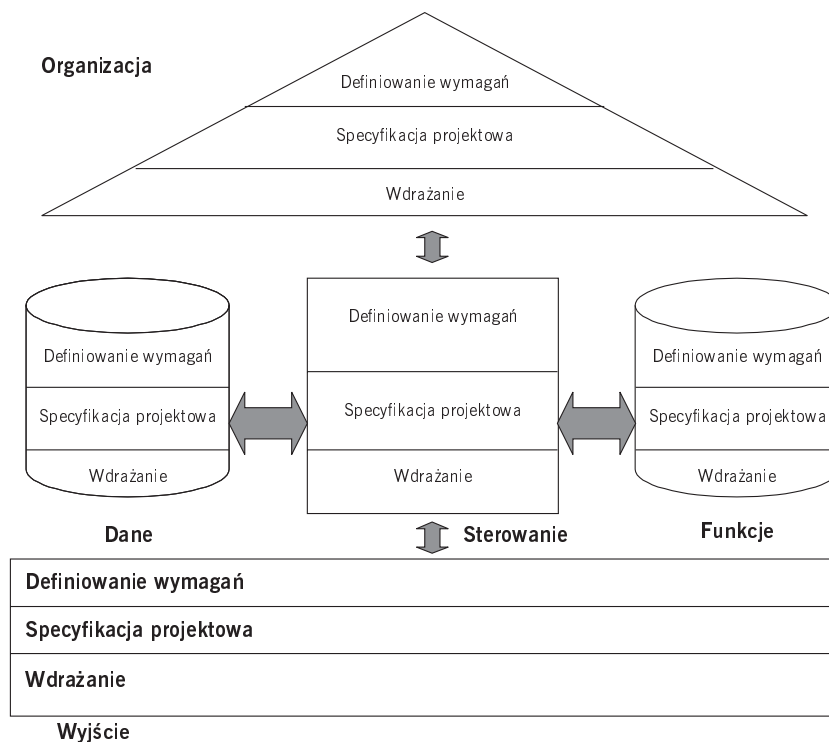
Programem umożliwiającym wykorzystanie podejścia obiektowego i UML jest także program ARIS (*Architecture of Integrated Information Systems*) opracowany przez niemiecką firmę IDS Prof. Scheer. W architekturze ARIS wyróżnia się następujące punkty widzenia na modelowane procesy (por. rys. 2.):

- funkcji (ang. *function view*) — opis funkcji (transakcji) przedsiębiorstwa oraz związków między nimi w postaci hierarchicznej struktury powiązań;
- danych (ang. *data view*) — opis zdarzeń (*events*) i stanów (*states*), które odnoszą się do działalności przedsiębiorstwa;
- organizacji (ang. *organization view*) — opis jednostek organizacyjnych i zasobów ludzkich, zaangażowanych w procesy przedsiębiorstwa oraz relacje między nimi i ich struktury;
- sterowania (ang. *control view*) — opis powiązań pomiędzy trzema wymienionymi powyżej punktami widzenia. Ten opis obejmuje elementy wszystkich pozostałych punktów widzenia.

W najnowszej wersji ARIS wprowadzony został punkt widzenia: *output view*, przedstawiający charakterystykę wejść i wyjść dla systemów [Scheer, Abolhassan, Jost, Kirchmer, 2002]. W przypadku modelowania procesów gospodarczych mogą to być na przykład wyroby wytwarzane w procesie produkcji lub produkty dostarczane klientowi w wyniku realizacji zamówienia.

Każdy z punktów widzenia jest przedstawiany na trzech poziomach opisu, odpowiadających trzem fazom budowy systemu, wydzielonych według kryterium „bliskości” do technologii informacyjnej: definiowania wymagań, specyfikacji projektowej i wdrażania.

Rysunek 2. ilustruje tradycyjne podejście do modelowania w ARIS. Widoczne jest tu oddzielenie modelowania danych od modelowania funkcji. Są one oddzielnie modelowane, a następnie „wiązane” za pomocą bloku sterowania.

**Rys. 2.**

**Perspektywy modelowania dostępne w programie ARIS**

Źródło: [Scheer, Abolhassan, Jost, Kirchmer, 2002].

Interesującą propozycję połączenia ujęcia modelowania proponowanego w ARIS z podejściem obiektowym przedstawił W. Fliegner [2001], proponując i przedstawiając metodę powiązania UML z narzędziami opisu i modelowania procesów ARIS, a przede wszystkim podstawowymi diagramami używanymi do modelowania procesów gospodarczych w ARIS, jakimi są diagramy łańcuchów procesów sterowanych zdarzeniami (ang. *Event-Driven Process Chains* — EPC).

Obecnie koncepcje obiektowości włączono do ARIS w znacznie szerszym zakresie, o czym można się przekonać, przeglądając najnowszą wersję tego programu, dostępną także w wersji polskiej (*ARIS 6.02 PL Collaborative Suite*) [ARIS 6.02 PL..., 2002].

### 3.3. Program iGrafx Process

Jeszcze jednym z programów, w którym wprowadzono modelowanie obiektowe, jest program iGrafx Process firmy Micrografx (obecnie Corel). Został on opracowany z przeznaczeniem do modelowania i graficznej wizualizacji procesów gospodarczych (sporządzania tzw. map procesów), ich analizy, symula-

cji przebiegu oraz zestawiania raportów dotyczących różnych charakterystyk procesów: czasu, kosztów, wykorzystywania zasobów itd. [Lasek, Pęczkowski, 2001]. Program jest przeznaczony do wspomagania planowania i tworzenia organizacji pracy w przedsiębiorstwach przemysłowych, bankach, urzędach administracji. Ułatwia podejmowanie właściwych decyzji dotyczących czasu, jaki powinny zajmować poszczególne etapy procesu gospodarczego, harmonogramu realizowanych czynności, danych (wejścia) i wyników (wyjścia) czynności, dostępności i wykorzystania zasobów (ludzie, maszyny, urządzenia), kosztów przetwarzania transakcji. Możemy np. utworzyć „modelową” firmę, złożoną z takich działów jak: kontaktów z klientami, produkcyjnego, badawczego. Wprowadzenie czynności, danych personalnych, czasowych i finansowych pozwala badać, ile i jakich zasobów potrzeba do realizowania zadań i jak dużo czasu to zajmuje. Posługując się programem iGrafx Process, możemy znaleźć, co da się ulepszyć w organizacji pracy, odkryć błędy, zanim pojawią się one naprawdę, obliczyć i prześledzić statystyki czasu pracy, kosztów oraz wykorzystania zasobów. Symulacja jest tu podstawową metodą umożliwiającą analizę i dostarczającą przesłanek do usprawnienia procesów. Pozwala wykryć nieprawidłowości w projekcie realizacji procesu. Przykładem nieprawidłowej organizacji procesu może być powstawanie kolejek prowadzących do zablokowania procesu. Wtedy transakcje będą coraz dłużej oczekiwać na przetworzenie, bo przychodzą zbyt często, aby zostały wykonane na czas przez pracownika przy założonym czasie wykonania jednej czynności. W takiej sytuacji musimy zmienić strukturę organizacyjną lub parametry procesu i wybrać inną alokację zasobów lub inną kolejność realizacji czynności.

Modelowanie procesów oparte na obiektowym podejściu umożliwia poprawę jakości procesów, początkowo poprzez eliminację błędów, a w dalszej kolejności poprzez doskonalenie przebiegu procesów, co ma na celu doprowadzenie do zmniejszenia kosztów operacyjnych, podniesienia poziomu wydajności, podwyższenia marży zysku, skrócenia cyklu produkcyjnego, ograniczenia zapasów, sprawniejszego przebiegu transakcji handlowych. Może zachodzić potrzeba przebudowy procesów zgodnie z założeniami BPR (*Business Process Reengineering*). Programy są rozbudowywane o nowoczesne metody doskonalenia procesów. Przykładem może być wspomaganie metody *Six Sigma* [DOE. *Warsztaty...*, 2002]. Aby zapewnić wspomaganie metody *Six Sigma* program iGrafx Process został rozbudowany poprzez połączenie wykorzystania możliwości programu iGrafx Process z możliwościami pakietu statystycznego Minitab. Opracowano program iGrafx Process 2003 for Six Sigma, włączając także możliwość modelowania obiektowego (język UML).

#### 4. Metoda *Six Sigma* doskonalenia jakości procesów gospodarczych

Najogólniej, metoda *Six Sigma* polega na przeprowadzaniu analizy procesów za pomocą metod statystycznych, a następnie ich optymalizacji na podstawie wyników tej analizy. Pozwala połączyć metody graficznej wizualizacji, analizy i projektowania procesów gospodarczych i systemów informatycznych z rozbudowaną analizą statystyczną. Dokładniej, *Six Sigma* wykorzystuje metody gromadzenia danych i narzędzia analizy statystycznej, aby wykryć przyczyny powstawania wad (braków) procesu i znaleźć sposoby ich wyeliminowania [DOE. *Warsztaty...*, 2002]. Celem jest poprawa jakości, która przyniesie korzyści klientom i firmie.

Zakłada się, że poprawa jakości nastąpi początkowo przez eliminację błędów, a w dalszej kolejności poprzez doskonalenie przebiegu procesów gospodarczych, co ma prowadzić do zmniejszenia kosztów operacyjnych, podniesienia poziomu wydajności, podwyższenia marży zysku, skrócenia cyklu produkcyjnego, ograniczenia zapasów, sprawniejszego przebiegu transakcji handlowych.

Występujące w nazwie metody określenie *Sigma* jest odwołaniem do odchylenia standardowego w rozkładzie normalnym, oznaczanego zwyczajowo grecką literą sigma ( $\sigma$ ). *Six Sigma* — sześć sigma, to 3,4 wadliwych produktów na 1 milion wykonanych i pożądaną poziom jakości do osiągnięcia przez firmę.

Wiele zjawisk w przyrodzie, a także w procesach gospodarczych wykazuje zgodność z rozkładem normalnym. Wynika to z faktu, że jeżeli na wynik wpływa bardzo dużo różnokierunkowych i niezależnych czynników, to ich łączny efekt można wyrazić za pomocą krzywej Gaussa. Można to udowodnić matematycznie — mówi o tym centralne twierdzenie graniczne w rachunku prawdopodobieństwa.

W związku z powyższym można przyjąć, że ustalony parametr procesu produkcji (np. średnica produkowanej śrubki, długość automatycznie ciętego drutu, waga pakowanej torebki towaru) jest zmienną losową o rozkładzie normalnym. Oznacza to, że jeżeli proces technologiczny przewiduje produkowanie np. śrubek o średnicy 10 mm, to kolejne egzemplarze nie będą miały dokładnie 10 mm każdy, ale ich średnice będą odchylać się od tej liczby w górę albo w dół. Mówimy, że proces jest prawidłowo wyregulowany, jeżeli wartość oczekiwana procesu wyniesie 10 mm (wtedy wartość średnia dużej liczby wyprodukowanych śrubek wyniesie z dużą dokładnością 10 mm). Średnica poszczególnych śrubek (odchyłka od normy wynoszącej 10 mm) będzie zależeć od odchylenia standardowego procesu ( $\sigma$ ).

Rozkład normalny ma tę własność, że 99,73% wartości odchylających się od wartości oczekiwanej mieści się w przedziale  $(-3\sigma; +3\sigma)$ . Jeżeli ustalimy normę tak, że odchylenia mieszczące się w tym przedziale uznamy za zgodne z normą, a elementy o parametrze wykraczające poza ten przedział uznamy za

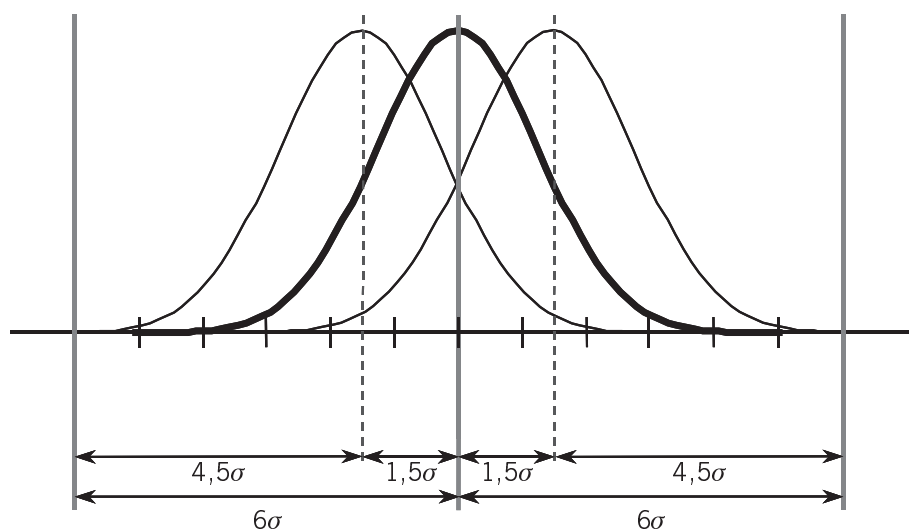


braki, to liczba braków będzie wynosić ok. 2700 na milion sztuk (2700 ppm — *parts per million*). Mówimy wtedy o jakości na poziomie 3 sigma.

Możemy postąpić odwrotnie, tzn. ustalić jakąś normę (przedział, w którym powinien się mieścić parametr), i zadbać o to, żeby proces miał jakość na poziomie 3 sigma. Jeżeli nie uzyskujemy takiej jakości, to należy poprawić technologię i organizację produkcji.

Liczba 2700 błędów na 1 milion, to jest jednak dużo błędów. Oznacza to, na przykład, że 2700 przesyłek pocztowych na 1 milion nie dochodzi do adresata. Aby zmniejszyć liczbę błędów, stworzono metodologię zakładającą jakość na poziomie 6 sigma. Jest to wysokie wymaganie — 6 sigma, to jest 2 razy więcej niż 3 sigma. Zgodnie z własnościami rozkładu normalnego mielibyśmy wtedy 0,002 braków na milion egzemplarzy (czyli ponad milion razy mniej niż w jakości 3 sigma). Jest to liczba niewiarygodnie mała, np. wytwarzając 1 sztukę produktu na sekundę, przeciętnie trzeba by czekać 15 lat na pierwszy brak. Osiągnięcie tak dobrej jakości produkcji jest w praktyce niemożliwe.

W metodologii tradycyjnej (3 sigma) uczyniono założenie, że proces technologiczny jest doskonale stabilny, tzn. wartość oczekiwana nie zmienia się w czasie. W rzeczywistości należy przyjąć, że jest to nierealne, i że nie tylko produkowane egzemplarze odchylają się od wzorca (10 mm), ale również parametr rozkładu normalnego (wartość oczekiwana) odchyła się od wzorca w procesie produkcji. Biorąc pod uwagę doświadczenia wynikające z praktyki w produkcji (głównie w przemyśle elektronicznym), przyjęto, że przesunięcia wartości oczekiwanej mieszczą się w przedziale  $\pm 1,5\sigma$  [DOE. *Warsztaty...*, 2002].



**Rys. 3.**

Zmiany parametrów procesów wokół wartości średniej uwidocznione po długim czasie

Źródło: A. Nowarski, P. Gawęd, *Wprowadzenie do SIX Sigma*, w: *DOE. Warsztaty: „Projektowanie eksperymentu za pomocą iGrafx System”*, mat. z konferencji „Six Sigma to bardzo proste”, KEMA, 18-19 października 2002.

W metodologii *Six Sigma* uwzględnia się fluktuacje wartości oczekiwanej i jeżeli będziemy mieścić się w normie  $6\sigma$  (6 odchyłeń standardowych) każdego z rozkładów normalnych, to uwzględniając fluktuację wartości oczekiwanej o  $\pm 1,5\sigma$  będziemy się mieścić w przedziale  $\pm 4,5\sigma$  dla wzorcowego rozkładu normalnego (rys. 3.). To wskazuje na możliwość utrzymywania się w normie przy założeniu  $\pm 4,5\sigma$ , czyli 3,4 braków na milion egzemplarzy.

Oprogramowanie *iGrafx Process 2000 for Six Sigma* umożliwia rozszerzenie modelowania i analizy procesów gospodarczych, jakie daje program *iGrafx Process 2000* o analizy statystyczne, dzięki zapewnieniu współpracy pakietu *iGrafx Process* z pakietem statystycznym *Minitab*.

Program *iGrafx Process for Six Sigma* umożliwia m.in.:

- znalezienie rozkładu statystycznego zmiennej dopasowanego do danych, charakteryzujących przebieg procesu, np. czas obsługi zlecenia klienta, czas dostawy produktu do klienta, koszty transportu; taki rozkład zmiennej jest traktowany jako wyrażenie (ang. *expression*), wykorzystywane w przeprowadzaniu symulacji przebiegu procesów;
- automatyczne przeprowadzanie eksperymentów przy wskazanych przez nas założeniach, dotyczących czynników decydujących o przebiegu procesu, np. sposobów realizacji zamówień klientów, stosowanych środków transportu, sposobu świadczenia usług: ekspresowego lub standardowego (ang. *factors*). W zależności od przyjętych założeń obliczane są mierniki wydajności lub jakości realizacji procesów (tzw. *responses*). Miernikami takimi mogą być, na przykład, koszty realizacji czynności lub całych „ścieżek” przebiegu procesów lub też stopień wykorzystania zasobów (ludzi, maszyn i urządzeń). Mogą też być zdefiniowane i zastosowane mierniki bardziej specyficzne dla metodologii *Six Sigma*, takie jak wskaźnik wydajności przejściowej (*Throughput Yield*), określający prawdopodobieństwo, że dany produkt lub usługa przejdzie przez dany etap procesu, nie wykazując żadnych wad, wskaźnik wydajności mierzonej w toku (*Rolled Throughput Yield*), przedstawiający prawdopodobieństwo, że produkt lub usługa przejdzie przez cały proces, nie wykazując wad, znormalizowany wskaźnik wydajności (*Normalized Yield*), będący przeciętnym wskaźnikiem wydajności przejściowej, obliczany przy uwzględnieniu wszystkich etapów procesu, tj. średniej wydajności przejściowej przypadającej na etap procesu. Po realizacji wskazanej przez użytkownika liczby eksperymentów wyniki mogą być poddane szczegółowej analizie statystycznej za pomocą procedur pakietu *Minitab*;
- przekazanie („wyeksportowanie”) wyników symulacji dla poszczególnych transakcji lub wyników eksperymentów do programu statystycznego *Minitab* lub arkusza kalkulacyjnego *Excel* w celu przeprowadzenia interesujących nas specjalistycznych analiz;
- przesłanie („wyeksportowanie”) raportów uzyskanych za pomocą programu *iGrafx* do zewnętrznych aplikacji w celu przeprowadzania dalszych analiz;

- przeglądanie i analizę przebiegu wybranej ścieżki lub ścieżek na mapie procesu w celu ich przededefiniowania lub tylko zmodyfikowania; analiza przebiegu ścieżek realizacji procesu jest dostępna w specjalnym oknie dialogowym *Process Analyzer*.

## 5. Podstawowe typy diagramów używane w modelowaniu obiektowym

Diagramy UML umożliwiające modelowanie systemów z różnych punktów widzenia (perspektyw) są omawiane m.in. w [Bluemke, 2001; Fowler, Scott, 2002; *Rational Rose...*, 2001; Bennett, McRobb, Farmer, 2002, Subieta 1999(a)]. Ilustrują je rysunki: rysunek 4. — diagramy udostępniane przez program *Rational Rose Enterprise Edition*, rysunek 5. — diagramy sporządzane za pomocą programu *ARIS*, rysunek 6. — diagramy uzyskiwane podczas posługiwania się programem iGrafx.

Diagramy przypadków użycia (*use case diagrams*) przedstawiają wykorzystanie systemu widziane z perspektywy zewnętrznej (aktorów), np. użytkowników lub systemów zewnętrznych. Ukazują one funkcjonalność systemu i interakcje zachodzące między systemem a światem zewnętrznym. Na przedstawionym diagramie (rys. 4.) wykorzystywane są symbole oznaczające aktorów oraz przypadki użycia. Symbole są połączone strzałkami ilustrującymi powiązania. Napisy «*uses*» («zawiera») przy niektórych strzałkach oznaczają, że prowadzą one do takiego przypadku użycia, który może występować wielokrotnie. Napis «*extends*» («rozszerza») przy niektórych strzałkach na diagramie przypadku użycia oznacza taki przypadek użycia, który (w pewnych sytuacjach) rozszerza wskazany strzałką przypadek użycia.

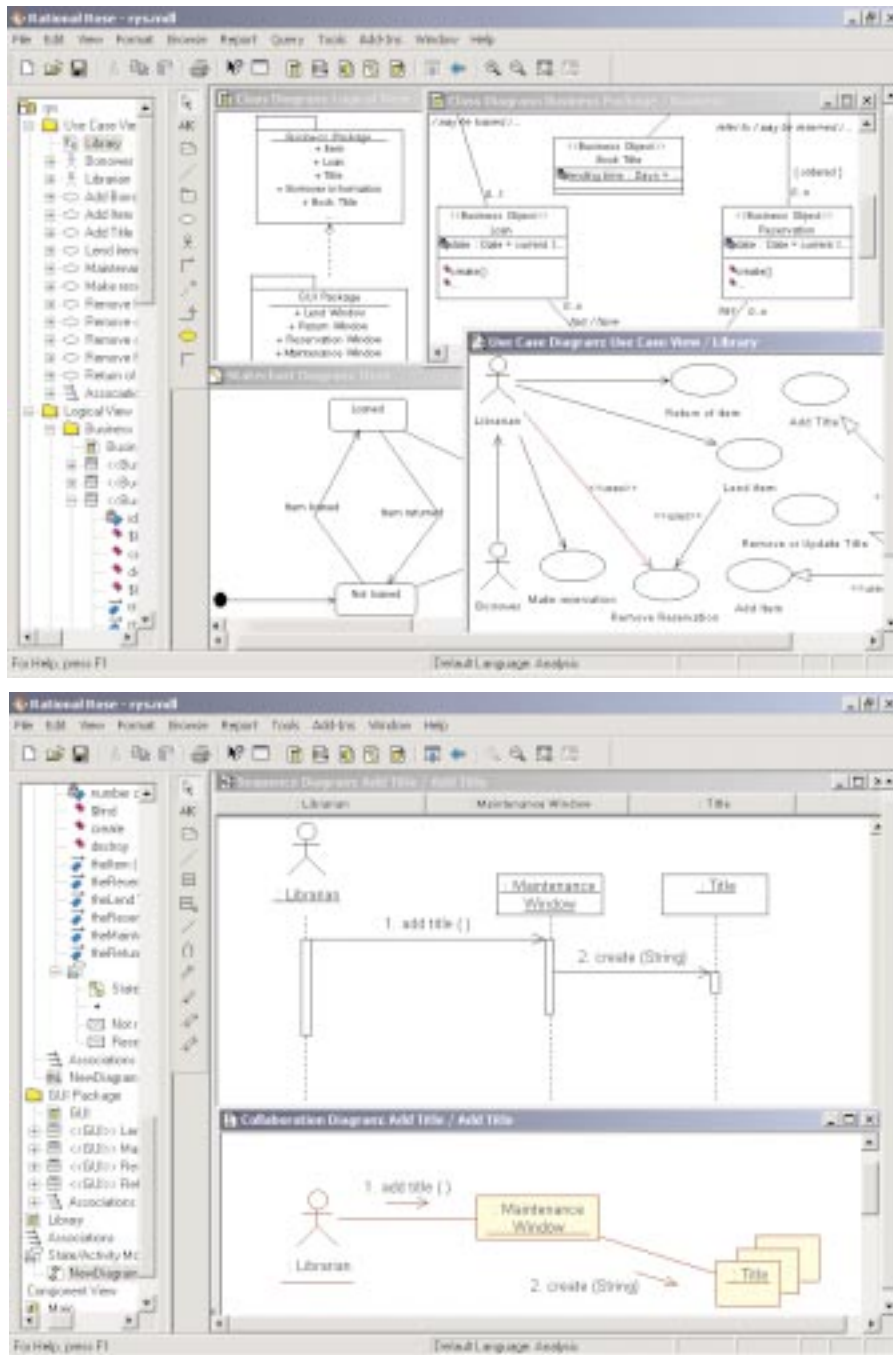
Podczas analizy diagramy przypadków użycia mogą być wykorzystane do przedstawienia wymagań i zrozumienia tego, jak powinien pracować system. W fazie projektowania diagramy przypadków użycia umożliwiają precyzyjne i jednoznaczne zdefiniowanie zachowania systemu po implementacji.

Diagramy klas (*class diagrams*) prezentują strukturę, statyczną wizję systemu.

Diagramy klas przedstawiają klasy obiektów, nazwy i typy atrybutów, nazwy metod, związki asocjacji i agregacji między obiektami, licznosci (*cardinalities*) tych związków oraz relacje dziedziczenia (generalizacji lub specjalizacji) między klasami.

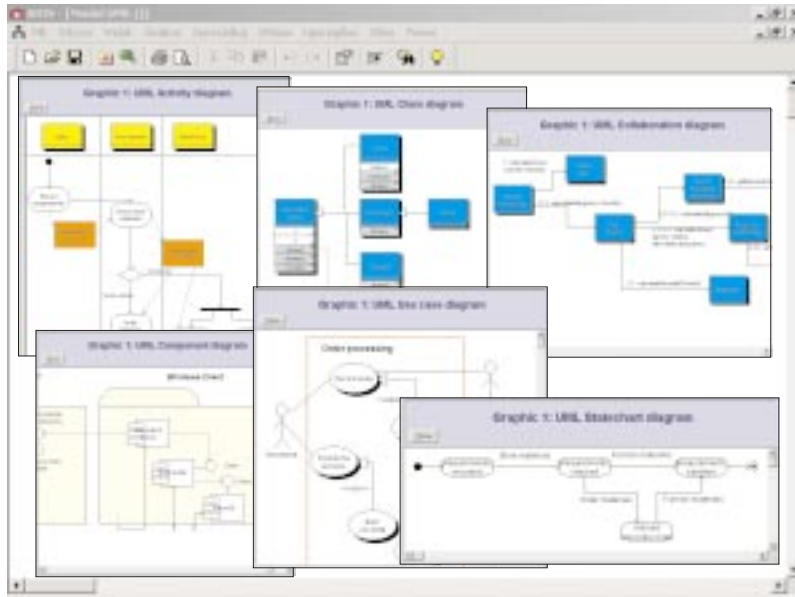
Klasy przedstawiane są w postaci prostokątów z opisami, jak na rysunku 7.

Pomiędzy klasami obiektów wprowadzane są związki zwane asocjacjami. Szczególnym przypadkiem asocjacji jest agregacja, przedstawiająca relację całość–część pomiędzy obiektami.

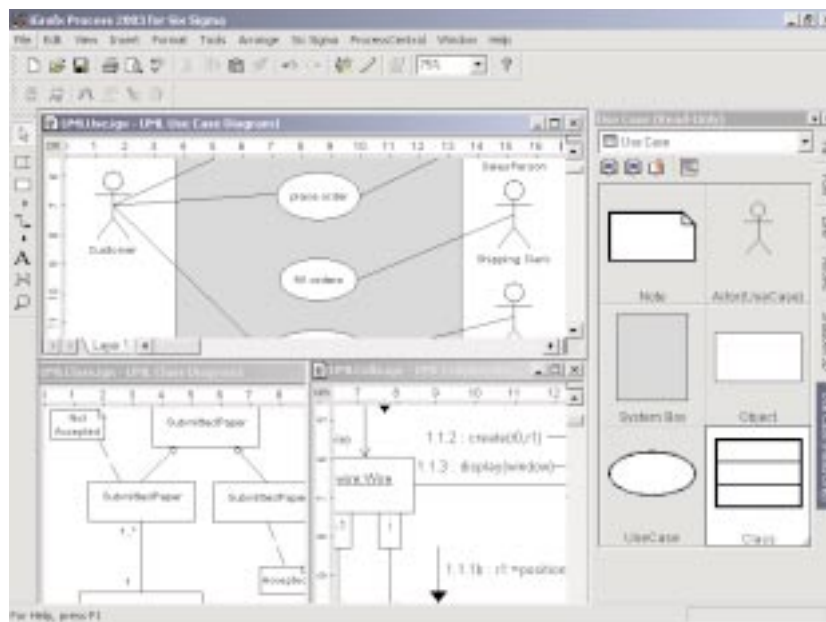


**Rys. 4.**

Przykłady diagramów modelowania obiektowego programu Rational Rose Enterprise Edition  
Źródło: przykłady dydaktyczne udostępniane z programem.



**Rys. 5.** Przykłady diagramów modelowania obiektowego programu ARIS  
Źródło: przykłady dydaktyczne udostępniane z programem.



**Rys. 6.** Przykłady diagramów modelowania obiektowego programu iGrafx Process 2003 for Six Sigma  
Źródło: przykłady dydaktyczne udostępniane z programem.



## Rys. 7.

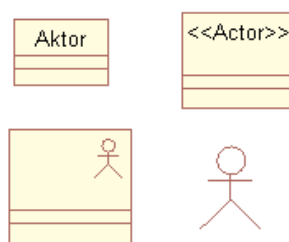
### Przedstawianie klas obiektów

Źródło: program Rational Rose Enterprise Edition.

Liczności przedstawiają liczby obiektów, które mogą być powiązane asocjacją. Liczności mogą być oznaczane w różny sposób, np.: 0..\* — zero lub więcej, \* — wiele, 1 — jeden, 0..1 — żaden lub jeden.

Na diagramach klas przedstawiane są relacje generalizacji, tj. relacje pomiędzy daną klasą i jej klasą nadrzędną. Podczas tworzenia diagramów generalizacja odbywa się poprzez utworzenie nowej klasy nadrzędnej z jednej lub więcej klas. Specjalizacja jest przeciwieństwem generalizacji i oznacza relację pomiędzy daną klasą i jej klasą podrzędną, a podczas budowania diagramów polega na utworzeniu nowej podklasy dla danej klasy.

Stereotyp (*stereotype*) jest rodzajem obiektu o określonej semantyce. W UML stereotypy mogą być oznaczane ciągami znaków zawartych w nawiasach: « » lub za pomocą ikon (por. rys. 8).



## Rys. 8.

### Przykład różnego sposobu przedstawiania stereotypów

Źródło: Program Rational Rose Enterprise Edition.

Diagram sekwencji (*sequence diagram*) ilustruje przebieg przesyłania komunikatów przez obiekty. Stanowi on ilustrację sekwencji zdarzeń. Obiekty lub klasy obiektów stanowią „nagłówki” kolumn na diagramie (zazwyczaj obiekty są przedstawiane w postaci prostokątów). Od każdego obiektu prowadzi linia pionowa przedstawiająca przebieg „życia” obiektu (w czasie). Strzałki poziome reprezentują komunikaty wysyłane przez obiekty do innych obiektów, przedstawiając kolejność zdarzeń. Miejsce na linii pionowej, od którego wychodzi strzałka (komunikat), wskazuje, kiedy został on wysłany do innego obiektu. Stąd wykres ilustruje kolejność wysyłania komunikatów (przebieg zdarzeń).

Diagramy sekwencji są związane z przypadkami użycia. Pokazują sekwencję zdarzeń, jaka musi nastąpić, aby został spełniony przypadek użycia dla realizacji określonej funkcji systemu. Są diagramami stosowanymi do analizy i projektowania systemu z perspektywy *Use Case View*.

Diagramy współdziałania (*collaboration diagrams*) przedstawiają przepływ komunikatów pomiędzy obiektami. Współdziałanie pomiędzy obiektami obejmuje dwa aspekty:

- statyczną strukturę obiektów (obejmując związki, atrybuty, operacje) — tzw. kontekst współdziałania;
- sekwencję komunikatów wymienianych pomiędzy obiektami do realizacji określonego zadania.

Diagram współdziałania może być tworzony dla wybranych typów obiektów, dla określonych operacji lub dla określonych przypadków użycia.

Współdziałanie może dotyczyć skutków, jakie powoduje dany fragment programu w środowisku zewnętrznym lub wewnętrznej implementacji obiektów i ich zachowania [Subieta, 1999(a); Subieta, 1999(b)].

Czas może być tylko pośrednio przedstawiany poprzez odpowiednią numerację komunikatów.

Diagramy stanów (*state diagrams; state-machine diagrams; state charts*) umożliwiają modelowanie zachowania poszczególnych klas obiektów lub obiektów. Przedstawiają kolejne stany, przez które przechodzi obiekt, zdarzenia, które powodują przejście z jednego stanu lub działania do innego, oraz akcje, które wynikają ze zmiany stanu lub działania.

Stany są przedstawiane w postaci prostokątów o zaokrąglonych rogach. Za pomocą strzałek oznaczany jest przepływ sterowania (przejścia) między stanami. Strzałki są zaopatrzone odpowiednimi komentarzami, opisującymi przejścia stanów.

Diagramy czynności (*activity diagrams*) — zwane także diagramami aktywności, przedstawiają przepływ sterowania. Ich cechą charakterystyczną jest to, że umożliwiają w klarowny sposób zobrazowanie przebiegu równoległych procesów.

Tory (*swimlane diagrams*) przedstawiają podobnie jak diagramy czynności przepływ sterowania, ale elementy diagramu (czynności procesu) są rozmieszczone w pionowych strefach przedzielonych liniami, nazywanymi torami. Każdy tor diagramu reprezentuje klasę odpowiedzialną za określone czynności, np. wydział przedsiębiorstwa.

UML udostępnia dwa rodzaje diagramów fizycznych (zwanym też implementacyjnymi):

- diagramy składowych (*Component Diagrams*)
- diagramy wdrożenia (*Deployment Diagrams*)

Diagramy składowych przedstawiają strukturę komponentów oprogramowania, ich związki i przepływ komunikatów oraz interfejsy ze światem zewnętrznym. Na diagramach węzłami są komponenty, a strzałki wskazują przepływ informacji od klienta tej informacji do jej dostawcy. Mogą pokazy-

wać interfejsy poszczególnych komponentów. Komponent diagramu składowych może przedstawiać jeden fizyczny moduł kodu systemu.

Diagramy wdrożenia przedstawiają konfigurację elementów czasu wykonania: elementów sprzętowych, elementów oprogramowania, procesów oraz związanych z nimi obiektów. Elementy, które nie istnieją w trakcie czasu wykonania, nie są przedstawiane na tych diagramach. Węzłami diagramów są elementy czasu wykonania połączone przez linie odwzorowujące połączenia komunikacyjne. Węzeł na diagramie wdrożenia często reprezentuje pewien rodzaj jednostki obliczeniowej — w większości wypadków jest to element sprzętowy. Elementy diagramu są połączone strzałkami wskazującymi element wykorzystywany przez dany element.

Można rysować oddzielnie diagramy wdrożenia i diagramy składowych, ale można też umieszczać diagramy składowych na diagramach wdrożeń i w ten sposób pokazać, które komponenty działają w których węzłach diagramu wdrożenia.

## Podsumowanie

Analiza koncepcji, metod, technik i narzędzi podejścia obiektowego prowadzi do wniosku, że dzięki swoim cechom — takim jak klarowność używanych pojęć (terminów), łatwość odwzorowania rzeczywistości widzianej przez użytkownika w postaci sformalizowanych, czytelnych diagramów graficznych, standaryzacji — podejście to może być użyteczne nie tylko w przypadku wytwarzania oprogramowania, ale także modelowania procesów gospodarczych i ukierunkowanego na te procesy gospodarcze kształtowania architektury i infrastruktury informacyjnej w przedsiębiorstwach. W przypadku modelowania procesów gospodarczych możemy się posługiwać obiektami biznesowymi i klasami obiektów biznesowych i przedstawiać modele zrozumiałe dla przyszłych użytkowników oprogramowania, a jednocześnie wystarczająco czytelne i jednoznaczne dla potrzeb tworzenia tego oprogramowania. Dostosowanie opisu procesu do obowiązujących standardów obiektowych sprawia, że problemy dziedziny przedmiotowej są opisane w postaci zrozumiałej dla wszystkich współuczestników zespołu kształtującego procesy gospodarcze i ukierunkowane na te procesy systemy informatyczne. Zapewniona zostanie w ten sposób komunikacja między użytkownikami reprezentującymi określoną wiedzę dziedzinową a twórcami systemów informacyjnych.

Zastosowanie podejścia obiektowego w modelowaniu procesów gospodarczych może być bardzo korzystne:

- po zdefiniowaniu obiektów, klas obiektów, wykorzystywanie polimorfizmu i dziedziczenia pozwala „montować” całe modele procesów zamiast sporządzania oddzielnych diagramów;
- istnienie jednego standardu modelowania umożliwia podział dużego modelu na podmodele, które mogą być przygotowywane przez oddzielne zespoły;
- dzięki zastosowaniu zasady dekompozycji, zasady abstrakcji, zasady ponownego użycia — zgodnie z rozumieniem tych zasad przyjętym w podej-



ściu obiektowym, możliwe staje się skuteczne działanie w warunkach nawet bardzo wysokiego stopnia złożoności rozważanego problemu;

- tworzone diagramy stanowią dokumentację systemu, powstającą automatycznie podczas modelowania procesów;
- dzięki hermetyzacji zapewniony jest dostęp tylko do tych elementów, do których jest to niezbędne, co zapewnia większą klarowność przy tworzeniu i analizowaniu modeli,
- zastosowanie standardów obiektowości i UML umożliwia wielokrotne wykorzystanie tych samych modeli lub ich fragmentów;
- wykorzystanie dziedziczenia i polimorfizmu ułatwia rozbudowę tworzonych modeli, ponieważ można „dobudowywać” kolejne klasy z nowymi atrybutami i metodami;
- zastosowanie standardów pojęć ułatwia zmianę części modelu, tak aby dostosować go do potrzeb konkretnego użytkownika;
- zastosowanie standardów obiektowości (np. UML) ułatwia współpracę zespołów realizujących różne modele bez potrzeby stałego kontaktu;
- istnienie programów, takich jak Rational, ARIS czy iGrafx, ułatwia modelowanie, wprowadzanie zmian do modeli, symulacje, przeprowadzanie eksperymentów i optymalizację procesów;
- standardy i wymagana dyscyplina modelowania porządkują i ułatwiają zarządzanie modelowaniem nawet w przypadku procesów o bardzo wysokim stopniu złożoności;
- stosując podejście obiektowe w modelowaniu, możemy wykorzystać gotowe wzorcowe modele (modele referencyjne), co pozwala ograniczyć niepotrzebną powtarzalność prac (zastosować znaną z inżynierii zasadę ponownego użycia);
- czytelność i klarowność opisów modeli procesów pomaga zidentyfikować te procesy i czynności, które przyczyniają się do wzrostu wartości firmy, i wyeliminować procesy niewnoszące wartości lub zlecić ich realizację na zewnątrz (np. poprzez *outsourcing*).

Wszystkie zalety podejścia obiektowego nie powinny pozwolić nam zapomnieć, że nawet najdoskonalsza metodyka jedynie wspomaga modelowanie procesów gospodarczych. O powodzeniu decydują ludzie: ich inteligencja, doświadczenie, wiedza, intuicja, zaangażowanie. Stąd wynika coraz większe powodzenie metodyk stymulujących twórczość, zwanych metodykami społecznymi [Checkland, Scholes, 2001; Wrycza, 1999].

## Bibliografia

- ARIS 6.02 PL Collaborative Suite, dokumentacja programu, 2002, IDS Scheer Polska.  
Avison D., Fitzgerald G., 2003, *Information Systems Development: Methodologies, Techniques, and Tools*, McGraw-Hill.  
Bennett S., McRobb S., Farmer R., 2002, *Object-Oriented Systems Analysis and Design using UML*, McGraw-Hill.  
Bluemke I., 2001, *Inżynieria oprogramowania*, Wyższa Szkoła Informatyki Stosowanej i Zarządzania, Warszawa.

- Booch G., Rumbaugh J., Jacobson I., 2002, *UML przewodnik użytkownika*, Wydawnictwa Naukowo-Techniczne WN-T, Warszawa.
- Chabik J., 1999, *Obiektość — technologia, technika, praktyka*, Jesienna Szkoła PTI, Mragowo.
- Checkland P., Scholes J., 2001, *Soft Systems Methodology in Action*, John Wiley & Sons.
- Chmielarz W., 2000, *Zagadnienia analizy i projektowania informatycznych systemów wspomagających zarządzanie*, Wydawnictwa Naukowe Wydziału Zarządzania Uniwersytetu Warszawskiego, Warszawa.
- DOE. Warsztaty: „Projektowanie eksperymentu za pomocą iGrafx System”, 2002, mat. z konferencji „Six Sigma to bardzo proste”, KEMA, 18–19 października.
- Fliegner W., 2001, *Rozszerzenie standardu UML jako propozycja integracji podejścia procesowego i podejścia obiektowego*, w: B. Kubiak, A. Korowicki (red.), *HCI '2001. Interakcja człowiek–komputer w rekonstrukcji procesów gospodarczych i tworzeniu zintegrowanych systemów informacyjnych*, Gdańsk.
- Fowler M., Scott K., 2002, *UML w kropelce*, Oficyna Wydawnicza LTP, Warszawa.
- Lasek M., Pęczkowski M., 2001, *Controlling procesów gospodarczych — zastosowanie programu iGrafx Process 2000 firmy Micrografx*, „Controlling i rachunkowość zarządcza”, część I, nr 2, s. 13–16, część II, nr 3, s. 9–13.
- Lethbridge T. C., Laganière R., 2001, *Object-Oriented Software Engineering. Practical software development using UML and Java*, McGraw-Hill.
- Rational Rose, dokumentacja programu*, 2001, Rational Software Corporation.
- Scheer A.-W., Abolhassan F., Jost W., Kirchmer M., 2002, *Business Process Excellence. ARIS in Practice*, Springer, Berlin, Heidelberg, New York.
- Shalloway A., Trott J. R., 2002, *Projektowanie zorientowane obiektowo. Wzorce projektowe*, Wydawnictwo Helion, Gliwice.
- Sommerville I., 2001, *Software Engineering*, Addison-Wesley.
- Subieta K., 1999(a), *Słownik terminów z zakresu obiektości*, Akademicka Oficyna Wydawnicza PLJ, Warszawa.
- Subieta K., 1999(b), *Wprowadzenie do obiektowych metodyk projektowania i notacji UML (Unified Modeling Language)*, Jedenasta Górńska Szkoła PTI, Szczyrk.
- Wrycza S., 1999, *Analiza i projektowanie systemów informatycznych zarządzania. Metodyki, techniki, narzędzia*, PWN, Warszawa.
- <http://www.ids-scheer.com>
- <http://www.micrografx.com>
- <http://www.rational.com>

**A b s t r a c t** The object approach in the modelling of economic processes and information systems of enterprises

A

The object approach was worked out within the framework of programming engineering and is widely used by information scientists for the needs of designing, programming and application of programming systems. The purpose of the article is to present the object approach as an approach which can be highly useful for economists and successfully used by them in the modelling and optimization—with respect to the costs, time, the use of resources, business processes realized in industrial, service and insurance enterprises, banks, administration. In the article the advantages of using the object approach in the modelling of business processes were stressed. Three programmes were briefly discussed, available on WNE UW: Rational Rose Enterprise Edition, ARIS and iGrafx Process, which enable the application of the object approach for the modelling and optimization of business processes.