V Konferencja PLOUG
Zakopane
Październik 1999

CORBA - technologia stawiająca „i” przy Oracle 8i - jak ją okiełznać

Paweł Księżyk

BSC Polska

Autor

Producent Manager w BSC Polska. Certified Delphi C/S Developer. W przedstawicielstwie Borlanda na Polskę pracuje od daty jego powstania. Prowadzi szkolenia i zajmuje się pomocą techniczną do Delphi i C++Buildera ze szczególnym uwzględnieniem problematyki bazodanowej. Zajmuje się również technologiami rozproszonymi (w tym CORBA). Autor wielu artykułów na temat programowania. Uczestniczył jako prelegent w kilkunastu konferencjach.

Streszczenie

1. Architektura obiektów rozproszonych – na czym polega i dlaczego mielibyśmy jej używać.
2. Skąd się wzięła technologia CORBA.
3. Możliwości technologii CORBA – jakie są i jak je można wykorzystać.
4. Inprise VisiBroker – technologia, która siedzi w Oracle 8i.

VisiBroker firmy Borland - technologia w Oracle 8i

Firma Borland podpisała kontrakt licencyjny z Oracle Corporation. W ramach tego wielomilionowego kontraktu Oracle zdecydował się korzystać z VisiBroker'a firmy Inprise jako standardu dla technologii CORBA Object Request Broker (ORB). Do tej pory VisiBroker został już zintegrowany z Oracle8i(tm), Oracle(r) Application Server oraz wieloma innymi produktami firmy Oracle. *"Wieloplatformowe standardy korporacyjne takie jak Java oraz CORBA są bardzo przydatne dla użytkowników Oracle"* powiedział Matt Mosman, Wice-Prezes Oracle Corporation do spraw Rozwoju. *"Szukając najlepszego na rynku CORBA ORB do integracji z rozwiązaniami Oracle stwierdziliśmy, ze VisiBroker firmy Inprise jest produktem najbardziej zaawansowanym technologicznie pod względem skalowalności, pewności w działaniu oraz zgodności ze standardami CORBA.*"
Nowy Oracle 8i dzięki integracji w serwerze technologii CORBA i Java stanowi świetne narzędzie do tworzeni aplikacji nowej generacji – Internetowych i rozproszonych. Serwer Oracle 8i może działać zarówno jako serwer lub jako klient w technologii CORBA. Dzięki VisiBrokerowi – Borlandowej realizacji technologii CORBA możliwa jest integracja istniejących rozwiązań niezależnie od tego na jakiej platformie działają i w jakich narzędziach zostały stworzone.

Warto wiedzieć jak czym jest ta technologia i jak ją wykorzystać.

Rozproszone obiekty: Wyzwanie lat 90-tych

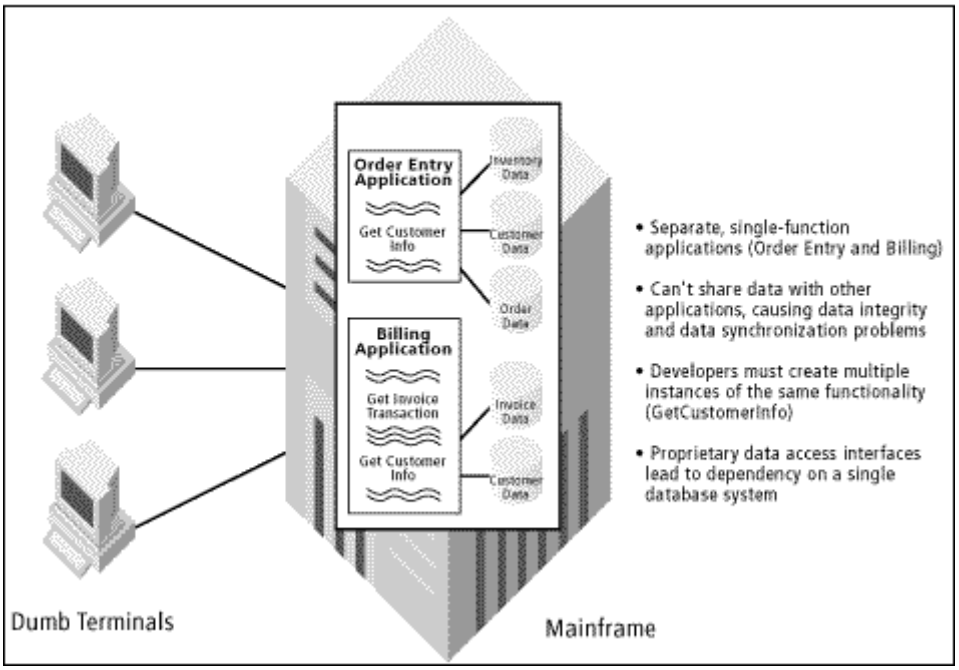
Stare przysłowie mówi, że "im więcej rzeczy się zmienia, tym więcej rzeczy pozostaje niezmiennych". Firmy w latach 90-tych stają przed tymi samymi wyzwaniami, z którymi miały do czynienia już od wielu dziesięcioleci. Przetrwanie lub sukces wymagają stałego udoskonalania obsługi klienta i jakości produktów. Jest to ciągła presja wywierana na granicę opłacalności. Technologia odgrywa coraz ważniejszą rolę w tworzeniu przewagi nad konkurencją. Cykle rozwoju w biznesie ciągle się skracają. Taka rzeczywistość przekłada się na coraz większy udział organizacji, w których kluczowym elementem jest Technologia Informacyjna (ang. *IT, Information Technology*). Zdolność do szybkiego reagowania ma krytyczne znaczenie dla sukcesu organizacji. Jednak przed działami IT krajobraz usiany jest przeszkodami, które utrudniają szybkie reagowanie.

Pod koniec lat 90-tych, większość głównych organizacji IT znajduje się pod dużą presją dostarczania większej wartości przy mniejszym koszcie. Większość z nich zmaga się z zarządzaniem złożonymi, heterogenicznymi środowiskami, które składają się rozmaitego sprzętu, oprogramowania, aplikacji, sieci i systemów baz danych. Wiele z nich musi integrować aplikacje mainframe zaprojektowane dziesięć lub dwadzieścia lat temu z najnowszymi technologiami internetowymi i klient/serwer. Popychane przez potrzebę dostarczania rozwiązań szybciej i efektywniej kosztowo, wiele organizacji IT było zmuszonych do przyjęcia taktycznych, krótkoterminowych podejść i w efekcie jest im coraz trudniej działać zgodnie z długoterminowym kierunkiem technicznego rozwoju przedsiębiorstwa.

Jak się tu znaleźliśmy?

Przez ostatnie 15 lat bardzo dużo się zmieniło w tym, jak projektujemy, programujemy i utrzymujemy korporacyjne systemy informacyjne. Okres ten rozpoczął się od monolitycznych systemów mainframe. Każdy z tych systemów zawierał całość swojej logiki prezentacji, biznesu i dostępu do danych. Nie były one w stanie dzielić swoich danych z innymi systemami, dlatego każdy z nich musiał przechowywać swoją prywatną kopię danych. Ponieważ różne systemy potrzebowały dostępu do tych samych danych, organizacje musiały przechowywać w wielu systemach nadmiarowe kopie danych.

Figure 1. The monolithic mainframe application architecture



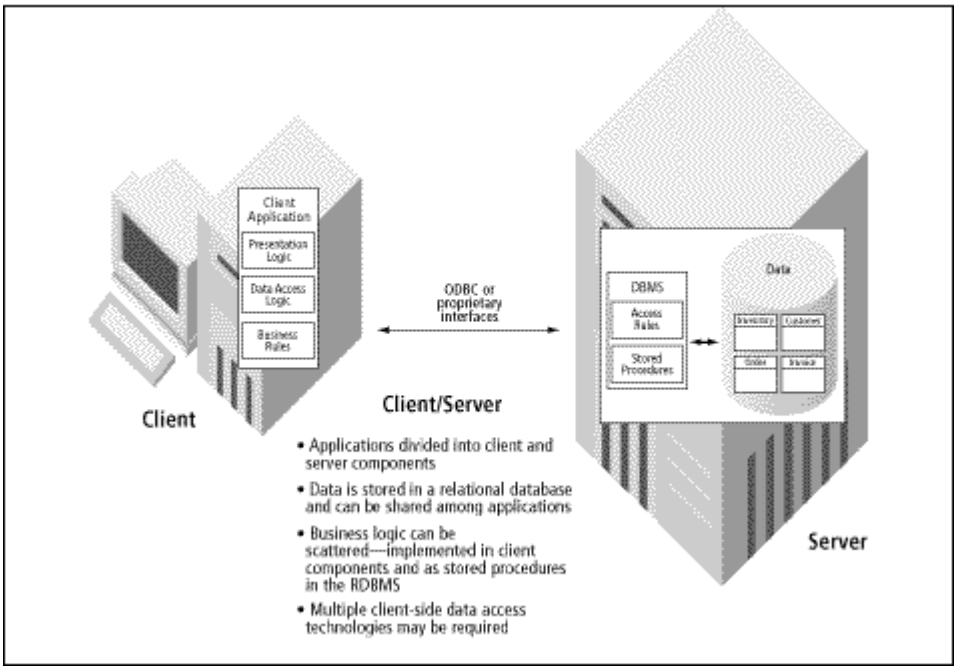
(Rysunek 1. Architektura monolitycznej aplikacji mainframe.)

Takie monolityczne aplikacje były niewydajne i kosztowne, dlatego musiały ustąpić pola technologii relacyjnych baz danych i modelowi klient/serwer. Możliwe dzięki konwergencji technologii – sieci, tanich komputerów PC, graficznych interfejsów użytkownika i relacyjnych baz danych – przetwarzanie klient/serwer obiecywało uproszczenie rozwijania i utrzymywania złożonych aplikacji poprzez wydzielenie ze zcentralizowanych, monolitycznych systemów komponentów, które można było łatwo programować i utrzymywać.

Aplikacje były podzielone na komponenty klienckie, które realizowały logikę prezentacji aplikacji i zawierały wiele z logiki roboczej (ang. *business logic*), natomiast komponenty serwerów zawierały logikę roboczą w postaci zachowanych procedur. Logika dostępu do danych była obsługiwana przez klienta albo przez serwer, zależnie od strategii implementacji. W efekcie wiele rozwiązań klient/serwer składało się po prostu z dwóch monolitycznych systemów, które kiedyś stanowiły całość. Dzisiaj nadal trudno jest budować, utrzymywać i rozszerzać aplikacje klient/serwer o znaczeniu krytycznym (ang. *mission critical*).

W tym czasie zespoły programistów musiały tworzyć tę samą funkcjonalność za każdym razem na nowo. Ponowne wykorzystanie kodu było trudne. Zwykle sprowadzało się do przekopiowania segmentu kodu, zmodyfikowania go, a następnie wdrożenia zmodyfikowanej kopii. W miarę upływu czasu coraz większa ilość podobnych modułów musiała być uaktualniana i utrzymywana oddzielnie. Zmiana w jednym module musiała być wprowadzana do podobnych modułów w całym przedsiębiorstwie. Ponieważ często okazywało się to całkowicie niezarządzalne, w konsekwencji do korporacyjnych systemów informacyjnych wkradały się funkcjonalne niespójności.

Figure 2. **The client/server application architecture**



(Rysunek 2. Architektura aplikacji klient/serwer.)

Rewolucja obiektów rozproszonych

Technologia obiektów rozproszonych fundamentalnie wszystko zmieniła. Razem z potężną infrastrukturą komunikacyjną, rozproszone obiekty rozbiły monolityczne aplikacje klient/serwer na samozarządzalne komponenty, które mogą ze sobą współdziałać pomiędzy różnymi sieciami i systemami operacyjnymi.

Komponentowy, rozproszony model przetwarzania obiektowego pozwolił organizacjom IT budować infrastrukturę, która łatwo adaptuje się do zachodzących zmian i odpowiada na wyzwania rynku. W wieku globalnej konkurencji i kurczących się nisz rynkowych, firmy, które potrafią inicjować gwałtowne zmiany – a nie tylko na nie reagować – są lepiej przygotowane do wykorzystania okazji i mają większą szansę odniesienia sukcesu.

Aplikacje rozproszone stanowią szansę stworzenia i utrzymania przewagi konkurencyjnej przez stworzenie elastycznej infrastruktury IT. Jednak wnoszą one ze sobą nowe wymagania. Aby móc działać w dzisiejszych, heterogenicznych środowiskach obliczeniowych, rozproszone aplikacje biznesowe muszą pracować na rozmaitych platformach sprzętowych i programowych. Muszą integrować starą technologię z nową i wykorzystywać istniejącą infrastrukturę. Co więcej aplikacje korporacyjne wymagają także możliwości wykraczających poza konwencjonalne przetwarzanie Webowe – skalowalności, wysokiej dostępności, łatwości administrowania, wysokiej wydajności i spójności danych.

Od 1989 roku, Object Management Group, konsorcjum złożone z wytwórców platform, niezależnych dostawców oprogramowania (ISV) i użytkowników końcowych pracowało nad specyfikacją architektury otwartej szyny programowej, poprzez którą obiekty napisane przez różnych wytwórców mogłyby współdziałać pomiędzy różnymi sieciami i systemami operacyjnymi. OMG to największe na świecie konsorcjum związane z oprogramowaniem, zrzeszające ponad 700 organizacji, a opracowana

specyfikacja cieszy się dużym poparciem społeczności wprowadzającej nowe standardy i przemysłu programistycznego. Organizacja standaryzacyjna International Standard Organization (ISO) zaaprobowała standardy OMG. Standardy OMG są również przyjęte przez X/Open jako część specyfikacji Wspólnego Środowiska Aplikacyjnego (ang. *CAE, Common Application Environment*).

Specyfikacja CORBA

Prace OMG skupiały się na stworzeniu specyfikacji Powszechnej Architektury Brokerów Żądań Obiektów (CORBA). Specyfikacja CORBA określa szynę programową, noszącą nazwę Brokera Żądań Obiektów (ang.*ORB, Object Request Broker*), zapewniającą infrastrukturę przetwarzania dla rozproszonych obiektów. Pozwala aplikacjom klienckim komunikować się ze zdalnymi obiektami, wywołując operacje statycznie lub dynamicznie. Pod koniec 1994, OMG zaaprobowało specyfikację CORBA 2.0, obejmującą protokół komunikacji pomiędzy ORB-ami pod nazwą protokołu Internet Inter-ORB Protocol (IIOP).

Protokół IIOP jest wydajny, skalowalny i zorientowany na transakcje. Pracuje na szczycie TCP/IP, nie wymaga żadnej specjalnej konfiguracji i szybko staje się standardem komunikacji pomiędzy rozproszonymi obiektami wykonującymi się w Internecie lub w korporacyjnych intranetach. Czołowi wytwórcy narzędzi internetowych, m.in. Netscape i Oracle, w pełni wykorzystują IIOP jako podstawę swojej przyszłej oferty produktów. Obiekty zgodne z CORBA 2.0 mogą ze sobą w pełni współpracować, ponieważ do komunikacji wykorzystują protokół IIOP.

Usługi CORBA

Szkielet ORB jest rozszerzany modularnymi, dodanymi usługami na poziomie systemu, które dopełniają funkcjonalność ORB i są gotowymi elementami do budowania aplikacji biznesowych. Konsorcjum OMG zdefiniowało zbiór powszechnych usług obiektowych, do którego zaliczają się:

- **usługa nazewnictwa** (naming service)pozwalająca obiektom odnajdować się wzajemnie po nazwie
- **usługa zdarzeń** (event service) pozwalająca obiektom być subskrybentem kanału zdarzeń i być powiadamianymi o określonych zdarzeniach
- **usługa transakcji** (transaction service) definiująca reguły transakcyjności, koordynująca dwufazowe zatwierdzanie operacji pomiędzy obiektami
- **usługa bezpieczeństwa** (security service) zapewniająca funkcje autentyfikacji, autoryzacji, szyfrowania i audytowania służące do chronienia cennych danych i kontrolowania dostępu użytkowników do aplikacji i usług.

Usługi CORBA zapewniają funkcjonalność, która ma krytyczne znaczenie dla wielu korporacyjnych aplikacji. Ponieważ programiści nie muszą implementować tych podstawowych funkcji w każdym systemie, dlatego mogą się skupić na implementacji swoich własnych aplikacji i logiki roboczej.

Zalety CORBY

Konsorcjum OMG i jego członkowie nakreśliło najbliższą przyszłość, w której obiekty programowe ze zdefiniowanymi interfejsami współpracują ze sobą w obrębie korporacyjnego intranetu i poprzez Internet. Płynące z tego korzyści dla programistów aplikacji i organizacji IT są bardzo istotne:

- **Wybór.** Architektura CORBA jest otwartym rozwiązaniem opartym na opublikowanej specyfikacji. Jest zaimplementowana i obsługiwana przez szeroką gamę sprzętu i platformy systemów operacyjnych.

- **Elastyczność "plug-and-play".** Obiekt programowy zgodny z architekturą CORBA posiada zdefiniowany interfejs. Wszelka komunikacja odbywa się poprzez ten interfejs. Zmiany w implementacji obiektu nie wpływają na inne obiekty, dopóki interfejs obiektu pozostaje taki sam. Programista może kodować z myślą o zdefiniowanym interfejsie wiedząc, że wprowadzane modyfikacje nie wpłyną na inne części rozproszonej aplikacji.

- **Koegzystencja z istniejącymi systemami.** Technologia ORB zabezpiecza inwestycje w istniejące systemy. Aplikacja mainframe, moduł lub punkt wejścia mogą być "opakowane" kodem C++ lub Java definiującym interfejs do istniejącego kodu. Stworzenie takiego opakowującego obiektu daje istniejącemu kodowi interfejs zgodny z architekturą CORBA, dzięki czemu może on współpracować z innymi obiektami w rozproszonym środowisku obliczeniowym.

- **Współpraca.** Obiekty programowe zgodne z CORBA komunikują się przy pomocy protokołu IIOP i mogą ze sobą w pełni współpracować, nawet jeżeli zostały opracowane przez różnych wytwórców, którzy wzajemnie nie mają żadnej wiedzy o swoich obiektach. Mosty programowe umożliwiają komunikację pomiędzy obiektami zgodnymi z CORBA i obiektami rozwiniętymi w technologii Microsoftu ActiveX/DCOM. Korporacyjne organizacje IT mogą wybrać ORB, usługi CORBA i obiekty programowe w oparciu o zapewnianą przez nie funkcjonalność – nawet jeżeli pochodzą od różnych producentów.

- **Przenośność.** Obiekty programowe zgodne ze standardem CORBA są przenośne. To znaczy, że obiekty zbudowane na jednej platformie mogą być wykorzystywane na każdej innej z obsługiwanych platform.

Eksplozja Internetu

W czasie kiedy specyfikacja CORBA była opracowywana i finalizowana, rozpoczęła się niebywała ekspansja Internetu i sieci World Wide Web, która nieprzerwanie trwa do dzisiaj.

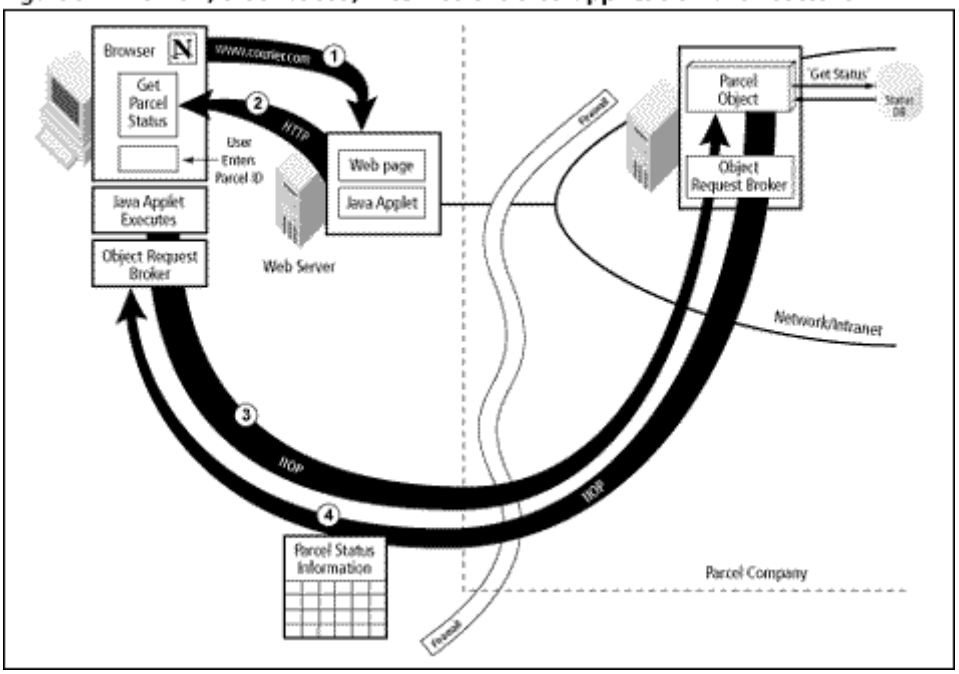
Wykraczając poza swoje korzenie tkwiące w agencjach rządowych i instytucjach edukacyjnych, Internet stał się najistotniejszym, nowym medium komunikacji pomiędzy organizacjami komercyjnymi, edukacyjnymi, rządowymi i osobami prywatnymi. Wzrost Internetu i korporacyjnych intranetów będzie się kontynuował w szybkim tempie przynajmniej do końca wieku, prowadząc do globalnej łączności, na skalę niespotykaną dotąd w historii przetwarzania komputerowego.

Nowy paradygmat aplikacji dla wieku Internetu

Równolegle rewolucje Internetu i rozproszonego przetwarzania obiektowego toczą się po zbiegających się ścieżkach. Jako szkielet komunikacyjny, Internet stanowi idealną platformę dla aplikacji opartych na rozproszonych obiektach i dlatego sprzyja ich rozwojowi. Jednocześnie technologia rozproszonych obiektów poprawia jakość aplikacji Webowych, zwiększając wartość Internetu i korporacyjnych intranetów.

Jak działa taka aplikacja? Rysunek 3 pokazuje przykład – rozproszoną, Webową aplikację do śledzenia paczek. Aplikacja pozwala użytkownikowi sprawdzić status dostarczenia przesyłki poprzez wprowadzenie numeru śledzonej paczki. Użytkownik wpisuje numer do formularza wyświetlanego przez aplet Javy, który wykonuje się w przeglądarce Weba użytkownika. Aplet Javy wykorzystuje Broker Żądań Obiektów do komunikowania się z obiektem programowym wykonującym się za firewallem firmy dostarczającej paczki; obiekt ten generuje zapytanie do firmowej bazy danych i zwraca informacje o statusie poprzez ORB do apletu Javy, który wyświetla informacje użytkownikowi.

Figure 3. **The new, distributed, Internet-enabled application architecture**



(Rysunek 3. Architektura nowej, rozproszonej, Internetowej aplikacji.)

Aplikacja śledzenia paczek działa następująco:

1. W przeglądarce WWW użytkownik wprowadza URL wskazujący na stronę znajdującą się na serwerze Weba firmy dostarczającej paczki znajdującym się na zewnątrz korporacyjnego firewalla.

2. Serwer Weba otrzymuje żądanie użytkownika i wykorzystuje HTTP do zwrócenia strony Weba do przeglądarki użytkownika. Strona Weba zawiera aplet Javy, który stanowi komponent kliencki rozproszonego systemu śledzenia paczek.

3. Do wykonującego się apletu użytkownik wprowadza numer paczki i klika przycisk "Pobierz Status Paczki" wyświetlany przez aplet Javy. Aplet wysyła komunikat IIOP poprzez Broker Żądań Obiektów do obiektu programowego po stronie serwera ("Paczka"), wywołując metodę obiektu "PobierzStatus" i przekazując do niej numer paczki wprowadzony przez użytkownika. Obiekt "Paczka" wysyła zapytanie o status paczki użytkownika do korporacyjnej bazy danych.

4. Po otrzymaniu wyników zapytania z bazy danych, obiekt "Paczka" zwraca wyniki do apletu Javy wykonującego się po stronie klienta poprzez Broker Żądań Obiektów. Aplet otrzymuje wyniki i wyświetla je w przeglądarce Weba użytkownika.

Klientem może być PC z systemem Windows, Macintosh, stacja robocza, komputer sieciowy, naręczny komputer osobisty, albo nawet przystawka do telewizora, taka jak WebTV. Obiekty serwerowe mogą być napisane w języku Java, C++, COBOL, SmallTalk lub innych i mogą zawierać nowy lub istniejący już wcześniej kod. Obiekty nie posiadają i nie potrzebują informacji o szczegółach swoich implementacji. Komunikują się między sobą tylko poprzez zdefiniowane interfejsy. W tle, przezroczyście dla użytkownika, Broker Żądań Obiektów zarządza komunikacją pomiędzy różnymi obiektami, z których składa się rozproszona aplikacja. Ponieważ klienci komunikują się z obiektami poprzez IIOP, dlatego mogą wywoływać funkcjonalność roboczą w sposób bezpośredni, bez przechodzenia przez oprogramowanie serwera Weba i opóźnień związanych z przetwarzaniem przez skrypt CGI każdego dostępu użytkownika.

Konsekwencje dla korporacyjnej organizacji IT

Nowy paradygmat aplikacji ma bardzo wiele do zaoferowania korporacyjnemu działowi IT. Organizacje, które szybko przyswoją sobie przetwarzanie w oparciu o obiekty rozproszone i wykorzystają Internet osiągną znaczącą przewagę nad swoimi konkurentami. Do korzyści tych można zaliczyć:

- Elastyczność mieszania i dopasowywania współdziałających ze sobą obiektów programowych pochodzących od różnych wytwórców
- Zcentralizowane zarządzanie i administracja obiektami programowymi
- Niższe koszty i krótsze okresy rozwijania oprogramowania wynikające ze stosowania obiektów wielokrotnego wykorzystania i uproszczenia integracji z istniejącym kodem
- Olbrzymie zmniejszenie kosztów nabywania, konfigurowania i utrzymywania systemów klienckich, spowodowane centralnym przechowywaniem aplikacji
- Stabilność powszechnie stosowanych standardów, takich jak CORBA i IIOP, zapewniająca współdziałanie, przenośność i szeroki wybór dostawców obiektów

CORBA jest po prostu specyfikacją programowej szyny. Możliwość wykorzystania aplikacji opartych na architekturze CORBA do zastosowań korporacyjnych zależy od implementacji ORB. Ze względu na swoją strukturę i kompletność, produkty VisiBroker są postrzegane jako wiodące rozwiązanie dla aplikacji rozproszonych. Po porównaniu różnych produktów ORB, dostawcy czołowych technologii na rynku – tacy jak Netscape, Oracle i Hitachi – wybrali produkty VisiBroker ze względu na ich implementację architektury CORBA i protokołu IIOP.

Inprise Application Server

Technologia ma dla działów IT dużych firm i korporacji wiele zalet. Wprowadzenie technologii przetwarzania rozproszonego wprowadza jednak wiele nowych wyzwań, do których należą problemy w zarządzaniu rozproszonym środowiskiem, konieczność poznania nowych usług i interfejsów. Borland stworzył więc kompleksowy zestaw zintegrowanych narzędzi, które odnoszą się do całego procesu tworzenia aplikacji – od tworzenia, poprzez rozpowszechnianie aż do zarządzania.

Z czego składa się Inprise Application Server:
Po stronie "front-endu" dostarczane są wizualne narzędzia:

- **JBuilder** To graficzne środowisko programistyczne - wiodące zintegrowane środowisko dla Javy - oferuje znajome wizualne środowisko do rozwijania wyrafinowanych rozproszonych aplikacji przy pomocy wskazywania i klikania. Pozwala tworzyć rozszerzenia aplikacji

korzystając z czarodziejów, którzy prowadząc krok po kroku dodają bezpieczeństwo i transakcje do rozproszonych aplikacji. Integracja z Delphi i C++ Builderem zapewnia całkowitą elastyczność wyboru narzędzi programistycznych.

- **AppCenter** Jest środowiskiem testowym i wdrożeniowym obejmującym możliwości konfigurowania różnych aspektów rozproszonej aplikacji. Oferuje zcentralizowaną, wizualną kontrolę nad rozproszoną aplikacją, pozwala administratorom stosować zautomatyzowane wykrywanie błędów i polityki naprawcze gwarantujące ciągłe działanie korporacyjnych aplikacji o krytycznym znaczeniu.

W swoim wnętrzu Inprise Application Server wykorzystuje następujące kluczowe technologie:

- **VisiBroker** Ta nagradzana technologia brokerów żądań obiektowych (ORB, Object Request Broker) - dla Javy i C++ - zapewnia skalowalność na poziomie korporacji, wysoką dostępność, zarządzanie połączeniami i wątkami oraz wydajność. VisiBroker dla Javy - pierwsza implementacja w Javie specyfikacji CORBA - zawiera GateKeeper'a, rozszerzenie, które zarządza komunikacją obiektową z serwera Weba.
- **VisiBroker Integrated Transaction Service (ITS)** Zintegrowana Usługa Transakcyjna VisiBroker ITS obsługuje transakcje dla aplikacji rozproszonych, zapewniając atomowość, spójność, izolację i trwałość wszystkich transakcji. Usługa VisiBroker ITS jest zgodna ze specyfikacją CORBA Object Transaction Service (OTS) i jest w pełni zgodna ze specyfikacją Java Transaction Service (JTS). VisiBroker ITS integruje się popularnymi bazami danych i mainframe'ami, zapewniając przezroczysty dostęp do wielu źródeł danych, wspierając zarówno środowiska XA, jak i nie-XA. Dane z istniejących systemów (takich jak monitory transakcji i oprogramowanie komunikacyjne) mogą być również odczytywane poprzez VisiBroker ITS, co zapewnia rzeczywistą łączność w obrębie całego przedsiębiorstwa.
- **VisiBroker SSL (Secure Sockets Layer)** Warstwa bezpiecznych gniazdek VisiBroker SSL zapewnia bezpieczne rozwiązanie zbudowane od podstaw z myślą o pełnych wykorzystywaniu Internetu i Weba, przy jednoczesnym zapewnianiu bezpiecznego, niezawodnego podejścia do wykonywania transakcji w rozproszonym środowisku. Stosowany model bezpieczeństwa opiera się na szyfrowaniu z kluczem publicznym (certyfikaty X509) i Warstwie Bezpiecznych Gniazdek (SSL) nad protokołem IIOP. Ten model bezpieczeństwa zapewnia autentyfikację i szyfrowanie, umożliwiając jednocześnie skalowanie na miarę korporacji.
- **Usługi Nazw i Zdarzeń VisiBrokera** Usługi te opierają się specyfikacji usług CORBA opracowanej przez Object Management Group (OMG) i oferują ułatwione zarządzanie obiektami i zdarzeniami w rozproszonej aplikacji. Inprise - lider w opartej na Javie technologii CORBA - zrealizował pierwszą implementację tych usług w Javie.
- **Serwer Webowy** Inprise Application Server posiada dołączony serwer Weba umożliwiający wdrażanie w Webie rozproszonych, wielowarstwowych aplikacji.

Co dalej

Przejście do wielowarstwowych serwerów aplikacji jest niezbędnym kolejnym krokiem w ewolucji rozproszonego przetwarzania obiektowego. W czasie dokonywania się tego przejścia, misją Inprise jest dostarczenie zintegrowanego środowiska upraszczającego rozwijanie, wdrażanie i zarządzanie wielowarstwowymi, rozproszonymi aplikacjami.

Inprise w postaci serwera Inprise Application Server oferuje potężne rozwiązanie problemu rozproszonych, wielowarstwowych aplikacji. Inprise Application Server zapewnia wszystko, co jest potrzebne przedsiębiorstwu: od standardowych przemysłowych protokołów, przez integrację z popularnymi zintegrowanymi środowiskami programistycznymi (IDE), aż po niezawodną infrastrukturę do przeprowadzania bezpiecznego, dwufazowego zatwierdzania transakcji, w których bierze udział wiele heterogenicznych (niejednorodnych) źródeł danych.

Poprzez zredukowanie programowania rozproszonych, wielowarstwowych obiektów serwerowych do operacji wskazywania i klikania, Inprise Application Server umożliwia składanie aplikacji z

wielokrotnego użytku komponentów logiki biznesowej znajdujących się w warstwie pośredniej. W efekcie szeregowi programiści będą w stanie rozwijać (a później wprowadzać modyfikacje) wyrafinowane aplikacje korporacyjne w rekordowo krótkim czasie.
Co więcej, dzięki zapewnieniu narzędzi do wdrażania i zarządzania - zbudowanych od podstaw z myślą o obsłudze rygorów rozproszonych aplikacji - Inprise Application Server daje korporacjom kluczowe narzędzia administracyjne. Korzystając z tych narzędzi administratorzy będą w stanie w łatwy sposób wdrażać rozproszone aplikacje, a następnie zarządzać nimi z jednego miejsca.
W konsekwencji przedsiębiorstwo będzie mogło szybko reagować na zmiany rynkowe i z powodzeniem konkurować na rosnącym rynku korporacyjnym.

Podsumowanie

W Oracle 8i znajdują się VisiBroker – Borlandowa implementacja technologii CORBA. Jest to dzięki temu bardzo dobre środowisko do uruchamiania aplikacji w postaci obiektów CORBA. Dzięki VisiBrokerowi możliwa jest integracja istniejących rozwiązań niezależnie od tego na jakiej platformie działają i w jakich narzędziach zostały stworzone.