

Krzysztof DOBOSZ

Politechnika Śląska, Instytut Informatyki

TWORZENIE APLIKACJI SIECIOWYCH W TECHNOLOGII JAVA 2 ENTERPRISE EDITION

Streszczenie. Publikacja prezentuje zastosowanie technologii J2EE w tworzeniu aplikacji sieciowych ze wskazaniem na jej wielowarstwowość i wykorzystanie poszczególnych składników. Omówione są również aktualne kierunki jej rozwoju ze szczególnym zwróceniem uwagi na ostatnio wprowadzone nowości.

Słowa kluczowe: Java, J2EE, serwisy internetowe

JAVA 2 ENTERPRISE EDITION TECHNOLOGY IN A NET APPLICATION CREATION

Summary. The paper presents the J2EE architecture in a net application creation, also indicates the multilevel construction and role of the most important parts. The main attention is directed on current directions of the J2EE development. These news are shortly described.

Keywords: Java, J2EE, web services

1. Wstęp

Java 2 Enterprise Edition (w skrócie J2EE) jest zestawem specyfikacji stworzonych przez firmę Sun Microsystems we współpracy z wieloma firmami obecnymi na światowym rynku oprogramowania. Dla tych specyfikacji stworzono wiele implementacji zarówno komercyjnych, jak i niekomercyjnych. Dla niektórych składników technologii J2EE firma Sun Microsystems udostępnia implementacje wzorcowe, które mają służyć ułatwieniu testowania innych implementacji, jednakże mogą być również wykorzystywane bezpośrednio.

Podstawą dla specyfikacji J2EE jest język Java. Język ten został stworzony z myślą o zastosowaniu w różnych urządzeniach, nie tylko w komputerach. Do największych zalet tego języka zaliczamy:

- 100% obiektowość,
- niezależność aplikacji od systemu operacyjnego,
- niezależność aplikacji od platformy sprzętowej,
- możliwość uruchamiania specjalizowanych aplikacji przez przeglądarki internetowe.

Wymagania użytkowników wykorzystujących oprogramowanie pracujące w sieci komputerowej są jednak znacznie większe. Sam język programowania, nawet jeśli posiada wiele zalet, nie jest w stanie ich zapewnić. Użytkownicy żądają od aplikacji sieciowych również takich cech, jak:

- bezpieczeństwo,
- skalowalność,
- możliwość współpracy z innymi aplikacjami,
- możliwość wykorzystania mocy i zasobów innych komputerów,
- zmniejszenie kosztów zakupu i wdrożenia aplikacji.

Projektowanie aplikacji sieciowych w standardzie J2EE stało się możliwe m.in. dzięki połączeniu w jedną całość kilku technologii informatycznych obejmujących różne obszary wykorzystywane przy projektowaniu i pisaniu aplikacji, począwszy od sposobu komunikacji z bazami danych, aż po metodologię tworzenia prezentacji.

2. Składniki

Na technologię J2EE składają się m.in.:

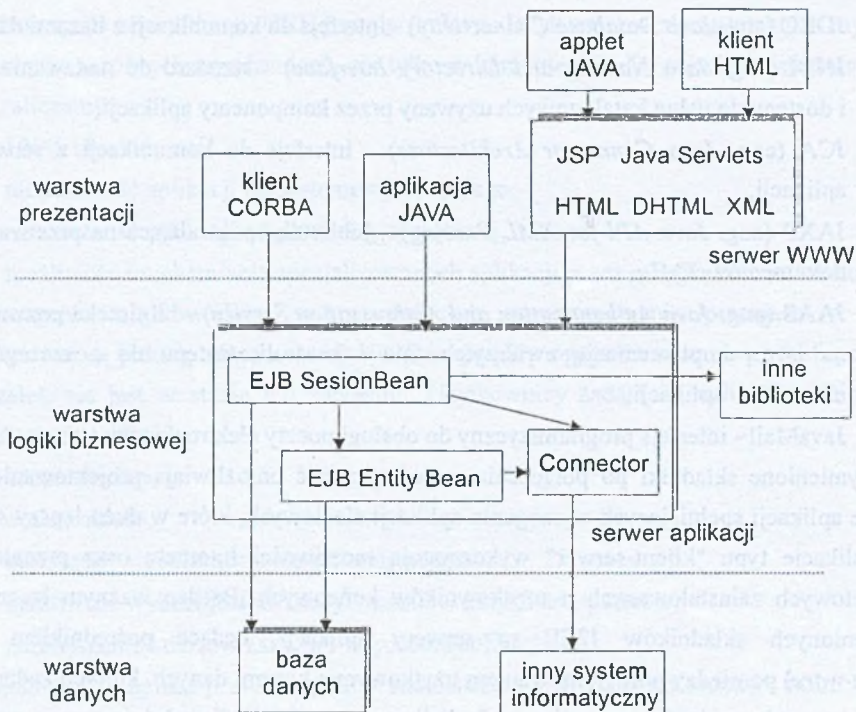
- język programowania Java wraz z maszyną wirtualną JVM,
- EJB (ang. *Enterprise Java Beans*) - technologia rozproszonych komponentów programowych,
- Java Servlets, JSP (ang. *Java Server Pages*) - technologia dynamicznej prezentacji danych na stronach WWW,
- RMI (ang. *Remote Method Invocation*) - protokół komunikacyjny,
- CORBA (ang. *Common Object Request Broker Architecture*) - architektura obiektów rozproszonych,
- JMS (ang. *Java Message Service*) - interfejs programistyczny do komunikacji asynchronicznej,
- JTA (ang. *Java Transaction API*) - interfejs do realizacji transakcji rozproszonych,

- JDBC (ang. *Java Database Connectivity*) - interfejs do komunikacji z bazami danych,
- JNDI (ang. *Java Naming and Directory Interface*) - standard do nadawania nazw i dostępu do usług katalogowych używany przez komponenty aplikacji,
- JCA (ang. *Java Connector Architecture*) - interfejs do komunikacji z serwerami aplikacji,
- JAXP (ang. *Java API for XML Parsing*) - biblioteka pozwalająca na przetwarzanie dokumentów XML,
- JAAS (ang. *Java Authentication and Authorization Service*) - biblioteka pozwalająca na łatwą implementację uwierzytelniania i kontroli dostępu do poszczególnych elementów aplikacji,
- JavaMail - interfejs programistyczny do obsługi poczty elektronicznej.

Wymienione składniki po połączeniu w jedną całość umożliwiają projektowanie oraz pisanie aplikacji spełniających wymagania aplikacji sieciowych, które w dużo lepszy sposób niż aplikacje typu "klient-serwer" wykorzystują możliwości Internetu oraz przeglądarek internetowych zainstalowanych u użytkowników końcowych. Bardzo ważnym łącznikiem wymienionych składników J2EE są serwery aplikacji, będące pośrednikiem (ang. *middle-ware*) pomiędzy oprogramowaniem użytkowym a bazami danych, których zadania nie są powiązane bezpośrednio z realizacją funkcji samej aplikacji. To właśnie serwer aplikacji zajmuje się rozwiązywaniem problemów związanych z bezpieczeństwem, skalowalnością, współpracą z serwerami internetowymi i innymi aplikacjami, współpracą z bazami danych oraz umożliwieniem wykorzystania zasobów sieci.

3. Architektura wielowarstwowa

J2EE powstała dla ułatwienia budowy, wdrażania i zarządzania skomplikowanymi systemami, zbudowanymi z użyciem architektury wielowarstwowej. Rozważane środowisko J2EE, którego schemat przedstawiono na rys.1, jest dostosowane do budowy wielowarstwowych systemów dostępu do baz danych w sieciach komputerowych. W architekturze tej możemy wskazać cztery podstawowe warstwy: klienta, komponentów WWW, komponentów EJB, oraz baz danych. Bez wątpienia najbardziej zaawansowanym elementem tej architektury są komponenty EJB, ponieważ posiadają one wbudowane mechanizmy zarządzania transakcjami, bezpieczeństwem i trwałością danych. Duże znaczenie posiada również zunifikowany interfejs do baz danych JDBC oraz protokoły komunikacji międzyobiektowej w postaci IIOP i RMI.



Rys. 1. Wielowarstwowa architektura J2EE

Fig. 1. Multilevel architecture J2EE

W architekturze zamieszczonej na rys.1 możemy wyróżnić kilka warstw składników realizujących odrębne funkcje.

A. Warstwa prezentacji

Klient może mieć dostęp do aplikacji poprzez osobną aplikację napisaną w Javie, instalowaną na komputerze (tzw. „gruby klient”), lub poprzez przeglądarkę internetową z dowolnego miejsca (tzw. „cienki klient”). Rozwiązanie z zastosowaniem „cienkiego klienta” umożliwia zminimalizowanie kosztów związanych z administracją i zarządzaniem systemem oraz wymaganymi umiejętnościami użytkowników, bowiem od użytkownika wymagana jest tylko znajomość obsługi przeglądarki internetowej. Użytkownik korzystający z przeglądarki internetowej komunikuje się z serwetem lub stroną JSP wykorzystując protokół HTTP. Servlety i strony JSP dynamicznie generują zawartość stron WWW. Dane pobierane potrzebne do wyświetlenia strony WWW brane są z warstwy środkowej (biznesowej) poprzez odpowiednie wywołania metod komponentów EJB.

B. Warstwa logiki biznesowej

Warstwa ta zawiera główną logikę biznesową aplikacji i jest ona odpowiedzialna za wykonywanie głównych jej zadań. Umożliwia ona dostęp do źródeł danych, integrację istniejącego systemu informatycznego przedsiębiorstwa z nowymi funkcjami. Warstwa oddziela system informatyczny przed warstwą prezentacji widoczną przez klienta. W warstwie tej umieszczone są komponenty EJB.

C. Warstwa danych

Warstwa danych jest źródłem danych dla aplikacji. Warstwę tę może stanowić relacyjna baza danych, system plików, system katalogowy czy obiektowa baza danych. Do warstwy tej zaliczany jest również dotychczasowy system informatyczny przedsiębiorstwa (o innej architekturze niż J2EE), z którego możemy pobierać potrzebne dane.

4. Nowe trendy

W ostatnim czasie pojawiła się na rynku kolejna edycja Java 2 Enterprise Edition v.1.4 wraz z głównym składnikiem, jakim jest pakiet JDK 1.4. Nie przyniosła ona znaczących zmian w samej technologii, jednakże można dostrzec kilka obecnie rozwijanych trendów związanych z tworzeniem aplikacji sieciowych.

4.1. Java Development Kit 1.4

Celem wersji JDK 1.2 i 1.3 było zapewnienie jak największej funkcjonalności - zawarto w nich dużo nowych bibliotek, z których na największą uwagę zasługiwał na pewno JFC. Dzięki tej funkcjonalności Java zaczęła być postrzegana jako język przydatny do tworzenia aplikacji klienckich. W nowej wersji 1.4 położono nacisk na zwiększenie stabilności i wydajności tworzonego w Javie oprogramowania.

Wersja 1.4 JDK ma w stosunku do poprzedniczek zwiększoną wydajność. 20-procentowy wzrost wydajności spowodowany jest krótszym ładowaniem klas, przyspieszeniem ładowania pełnych stron HTML. Nowa wersja przystosowana została także do systemów o większej skali - większej liczbie procesorów, zasobów i przetwarzanych danych. Średnio o 20 procent skraca się czas ładowania oprogramowania, przyspieszona zostanie interakcja z użytkownikiem. Wielokrotnie przyspieszone zostaną operacje wejścia - wyjścia i mapowania pamięci. Nowe JDK wyposażone jest w 64-bitową wirtualną maszynę.

4.2. XML

Standard XML związany jest z przenośnością danych. Biorąc pod uwagę fakt, że język Java powiązany jest z przenośnością kodu, mamy w ręku technologię pozwalającą na tworzenie elastycznych, niezależnych od platformy sprzętowej i programowej aplikacji internetowych. W Javie XML jest obsługiwany przez oddzielną bibliotekę Suna o nazwie JAXP (ang. *Java API for XML Parsing*). W jej skład wchodzi parsery SAX i DOM.

Sun rozwija także bibliotekę XML Data Biding. XDB oferuje API wyższego poziomu niż DOM i SAX, łącząc za pomocą XML Schema definicję dokumentu XML z jego przetwarzaniem i sprawdzaniem poprawności poszczególnych elementów drzewa. Obecnie najczęściej stosowanym mechanizmem sprawdzania poprawności plików XML jest DTD (ang. *Document Type Definition*). DTD było szybko stworzoną łatą na specyfikacji XML, aby umożliwić określanie struktury dokumentu w jawny sposób. Dla bardziej skomplikowanych wymagań jest to jednak niewystarczające.

Organizacja W3C (ang. *World Wide Web Consortium*) zaprezentowała o wiele bardziej dojrzałe i rozbudowane rozwiązanie - XML Schema. Programista może napisać zestaw klas, które będą reprezentować dokument XML. Za pomocą tych klas będzie można ładować, przetwarzać i sprawdzać poprawność składni dokumentu, tworząc obiekty Javy dla każdego węzła w drzewie XML. XDB umożliwia wygenerowanie klas automatycznie, jeśli tylko będzie mogła skorzystać z odpowiedniego schematu XML dokumentu.

4.3. Web Services

W najnowszej wersji J2EE oprócz aktualizacji pakietu JDK uaktualniono również specyfikacje EJB 2.1, JSP 1.3 i Java Servlet 2.4 API, Connector Architecture 1.5. Modyfikacji poddane zostały również usługi sieciowe (Web Services) oraz zarządzanie. Nowością w wersji 2.1 standardu EJB jest implementacja Web Services poprzez EJB.

Web Services jest warstwą izolującą, pozwalającą wywołać zdalną usługę bez znajomości szczegółów implementacyjnych. Ekspozuje tylko warstwę biznesową, udostępniając publiczne API dla współpracujących serwisów. Web Services możemy nazwać samodzielnymi aplikacjami, które mogą być opisane, opublikowane, zlokalizowane i wywołane w sieci internetowej. Każdy, kto dotrze do usługi, będzie mógł ją wywołać. Wywołującym może być proces biznesowy, który określi zapotrzebowanie na konkretną usługę oraz na interfejs komunikacyjny oparty na komunikatach. Aplikacja biznesowa odnajdzie w sieci usługę spełniającą zadane kryteria i wykona ją zdalnie, niezależnie od swojej platformy operacyjnej, formatu danych, języka programowania i implementacji.

Podstawową cechą Web Services jest zapewnienie możliwości komunikowania się i współpracy różnych aplikacji biznesowych. Jest to możliwe, ponieważ usługi sieciowe wykorzystują uzgodnione standardy struktury danych, przesyłania komunikatów, wyszukiwania usług i opisu interfejsów. Wymiana komunikatów między odbiorcą usług a dostawcą opiera się na protokole SOAP, którego komunikaty sformułowane są w języku XML, a ewentualne załączniki są binarne lub w postaci wywołań RPC. Wiadomości natomiast są przesyłane za pomocą standardowego protokołu internetowego HTTP.

Usługa sieciowa po odebraniu żądania wykonania interpretuje je i wywołuje odpowiednią, zaimplementowaną metodę (np. w postaci EJB) po stronie aplikacji biznesowej - faktycznego dostawcy usług. Odpowiedź musi zostać przetworzona na komunikat SOAP i wysłana do odbiorcy. Oczywiście odbiorca musi umieć go zinterpretować. Web Services mogą więc być informacyjną reprezentacją właściwych usług sprzedawanych przez firmę.

5. Uwagi końcowe

Na platformach zgodnych z J2EE opartych jest już bardzo dużo aplikacji i serwisów internetowych. Pracują one na różnych serwerach aplikacji, baz danych, platformach dostarczanych od różnych producentów. Przykładem może być największa księgarnia internetowa *Amazon.com*, która jest stworzona w architekturze J2EE i pracuje na serwerze aplikacji Bea WebLogic. Należy zauważyć, że oprócz Bea WebLogic istnieje wiele rozwiązań serwerów aplikacji, które są dostarczane od różnych producentów oprogramowania, np. IBM WebSphere, Oracle Oracle8i, Inprise Application Server czy Netscape Application Server.

Szczególnym zainteresowaniem w ostatnim czasie cieszy się metoda usług informacyjnych zwana Web Services. Web Services mogą pełnić funkcję informacyjnej reprezentacji właściwych usług sprzedawanych przez firmę. Uruchomienie sprzedaży nowej usługi poprzez Web Services na pewno skróci czas i obniży koszt opracowania i wdrożenia interfejsu sieciowego reprezentującego nowy produkt.

Przykładem zastosowania Web Services może być udostępnienie przez biuro podróży swoich aplikacji do rezerwowania hotelu, biletu lotniczego czy wynajmu samochodu w formie usług sieciowych. Klient biura będzie mógł podać agentowi elektronicznie szczegóły dotyczące wyjazdu wakacyjnego, a system agenta automatycznie znajdzie odpowiednią linię lotniczą, zarezerwuje hotel czy samochód i np. zaproponuje trasę zwiedzania zabytków rezerwując odpowiednie bilety wstępu. Po zakończeniu procesu system odeśle klientowi potwierdzenie dokonanych rezerwacji. Tak więc Web Services mogą wyręczać ludzi nie tylko w prostych operacjach, ale niejednokrotnie w podejmowaniu istotnych decyzji.

LITERATURA

1. Sun Microsystems. Witryna internetowa: <http://java.sun.com/j2ee/>
2. Sun Microsystems. The Java Web Services Tutorial. Witryna internetowa: <http://java.sun.com/webservices/docs/ea2/tutorial/>
3. Bond M., Haywood D., Law D., Longshaw A. Roxburgh P.: Teach Yourself J2EE in 21 Days. Sams Publishing, Indianapolis 2002.
4. Grudziński G.: JAVA czy .NET?. IV Jesienne Warsztaty Linuxowe. Zegrz 2001.

Recenzent: Dr hab. inż. Franciszek Marecki

Wpłynęło do Redakcji 28 marca 2003 r.

Abstract

The paper presents the J2EE architecture in a net application creation. At the beginning (chapter 1) the paper discusses J2EE and the properties are described. The base role of the Java language is indicated. Next (chapter 2) the most important parts of the J2EE architecture are numerated and shortly described. The most important part is an application server called "middle-ware". J2EE environment has a multilevel architecture (chapter 3), what is illustrated on the figure 1. The architecture consists of: presentation level ("thin client" – for internet viewer users, or "fat client" – for independent applications), business logic level (using EJB components for main tasks and access to a database) and data level (databases, file systems, folder systems). Next (chapter 4) current directions of the J2EE development are indicated. First new properties of the JDK 1.4 are characterized. Then chapter shows important role of the XML as a data description standard and pays attention to connection to the Java language as a powerful solution for data processing in the Internet. At last the Web Services are characterized as a new solution for cooperation between business applications. And at the end, chapter 5 contains conclusions.

Adres

Krzysztof DOBOSZ: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-101 Gliwice, Polska, kdobosz@star.iinf.polsl.gliwice.pl