

УДК 004.054

Чанцова Катерина Вікторівна

студент

НТУУ «КПІ ім. Ігоря Сікорського»

Чанцова Катерина Викторовна

студент

НТУУ «КПИ им. Игоря Сикорского»

Chantsova Kateryna Viktorivna

student

NTUU "Igor Sikorsky KPI"

ПОРІВНЯННЯ ТЕХНОЛОГІЙ ENTERPRISE JAVABENS ТА .NET
СРАВНЕНИЕ ТЕХНОЛОГИЙ ENTERPRISE JAVABENS И .NET
COMPARISON OF TECHNOLOGIES ENTERPRISE JAVABENS AND
.NET

Анотація: Розглянуто особливості технологій .NET та Enterprise JavaBeans, основні відмінності між ними.

Ключові слова: .NET, Enterprise JavaBeans, Java Enterprise Edition.

Аннотация: Рассмотрено особенности технологий .NET и Enterprise JavaBeans, основные отличия между ними.

Ключевые слова: .NET, Enterprise JavaBeans, Java Enterprise Edition.

Summary: The peculiarities of .NET and Enterprise JavaBeans technologies and the main differences between them are considered.

Key words: .NET, Enterprise JavaBeans, Java Enterprise Edition.

Microsoft .NET - самостійне повноцінне середовище розробки, яке серед багатьох інших можливостей включає і розробку розподілених додатків. Enterprise JavaBeans (EJB) - технологія створена з метою спрощення розробки розподілених додатків, є частиною Java Enterprise Edition (JEE). EJB, частина JEE, і Microsoft .NET надають набір стандартизованих модульних компонентів і сервісів. Маючи стандартні компоненти і сервіси в своєму розпорядженні, розробники можуть сконцентруватися на побудові бізнес-логіки додатка, замість того щоб програмувати більш фундаментальні речі. Використання візуальних інструментів значно спрощує розробку додатків.

Ключове відмінність між Java і .NET полягає в тому, що JEE є відкритим стандартом, який працює на декількох платформах, в той час як .NET є власністю Microsoft і працює виключно на Windows. Продукти JEE доступні від багатьох постачальників, в той час як технологія .NET доступна тільки від Microsoft.

Ще одна відмінність між JEE і .NET полягає в тому, що, використовуючи JEE, а відповідно і EJB, можна писати тільки на одній мові - Java. У той час як .NET підтримує кілька мов - основними з яких є Microsoft Visual Basic, C ++ і C #. Інші мови також можуть підтримуватися .NET, якщо вони переписані для запуску в середовищі .NET [1].

JEE і .NET надають бібліотеки компонентів. Бібліотека компонентів JEE включає API ядра Java, який включає Enterprise JavaBeans. .NET надає бібліотеки класів, які містять керовані компоненти .NET. .NET також використовує попереднє покоління COM + компонентів в операційній системі Windows. У моделі JEE окремі операції кожного рівня виконуються в окремих «контейнерах». Контейнер - це частина середовища виконання, яка послідовно виконує завдання - наприклад, вилучення даних і створення веб-сторінки. У .NET немає контейнерів, але є кілька шляхів, за якими операції можуть бути написані або оброблені.

JEE і .NET мають набір компонентів, сервісів і функцій, які забезпечують стандартний спосіб виконання таких завдань, як доступ до баз даних, створення сценаріїв веб-сторінок, обробка повідомлень і підключення до віддалених ресурсів [2]. Порівняння ключових функцій і сервісів JEE та .NET Наведено в таблиці нижче (табл. 1).

Таблиця 1

Порівняння ключових функцій і сервісів JEE та .NET

Service or feature	Microsoft .NET	JEE
Language	C#, VB.NET, C++.NET, more languages	Java
Operating System	Windows	Multiple
Runtime	CLR	JVM
Server Components	.NET, COM+	EJBs
Client/GUI Components	.NET class	JavaBeans
Web Server Scripting	ASP.NET	JSP/Servlet
Data Access	ADO.NET	JDBC
Persistent Objects	Business Entity Comp.	EJB Entity Beans
Message Queueing	Sys. Messaging, MSMQ	JMS on (MQSeries)
Asynchronous Invocation	COM+ QC	EJB Message Beans
Remoting	SOAP, HTTP, DCOM	RMI-over-IIOP
Naming	ADSI	JNDI
XML	System XML	JAXP
HTTP Engine	IIS	Application Servers from multiple vendors
Web Services Support	Built-in	Add-on

Як вже говорилося раніше, EJB не залежить від платформи, це є його перевагою в порівнянні з платформою .NET. Однак, незабаром це може змінитися, так як Microsoft працює над можливістю розробки за допомогою .NET на інших платформах.

У додатках .NET безпекою і пам'яттю управляє CLR, в додатках, побудованих за допомогою EJB, ресурсами, пам'яттю і безпекою управляє контейнер.

Компонент EJB оформлюється як невеликий набір класів і інтерфейсів на Java, а також має дескриптор розгортання (опис в певному форматі на основі XML конфігурації компонента в рамках контейнера). .NET - компоненти являють собою набір класів і конфігураційних файлів,

що грають роль дескрипторів розгортання і також представлених в деякому форматі на основі XML [3].

Для подальшого порівняння технологій .NET і EJB будуть розглянуті деякі особливості роботи з ними, пов'язані з такими параметрами, як зв'язок, іменування, процеси і синхронізація, цілісність, відмовостійкість, захист, робота з XML.

Зв'язок між компонентами, які працюють в різних процесах і на різних машинах, забезпечується в EJB, в основному, двома способами: синхронний зв'язок - за допомогою реалізації віддаленого виклику методів на Java (Java RMI), асинхронний - за допомогою служби повідомлень Java (Java message service , JMS). Зв'язок між компонентами в рамках .NET здійснюється за допомогою механізму Remoting, що реалізує як RMI, так і асинхронну передачу повідомлень, більш новою і досконалою технологією є WCF (Windows Communication Foundation).

Пошук ресурсів за іменами та ідентифікаторами і наборами їх властивостей в рамках EJB здійснюється за допомогою інтерфейсу JND (Java Naming and Directory Interface). В .NET в разі, якщо фізичне положення компонентів, з якими необхідно встановити зв'язок, відомо і є постійним, використовується локальна служба іменування, вбудована в середу. В іншому випадку, використовується Active Directory. Оскільки ця технологія з'явилася раніше, ніж середовище .NET, в рамках .NET була створена бібліотека адаптерів, що дозволяють використовувати функції Active Directory.

Розбиття програми на набір взаємодіючих процесів і потоків та керування ними здійснюється EJB-контейнером автоматично. Крім процесів і потоків, середа .NET підтримує так звані зони додатків (application domains). Вони, як і процеси, слугують агрегатами ресурсів, але управляються більш ефективними механізмами.

В рамках одного процесу може бути створено декілька зон додатків. Вони слугують додатковим елементом захисту .NET-додатків від ненавмисного впливу і дозволяють зберегти працездатність процесу у разі виникнення проблем в одному з його додатків. Окрім автоматично створюваних потоків і зон додатків, розробник може створювати свої власні потоки та зони додатків. Питання синхронізації потоків і передачі даних між зонами додатків можуть вирішуватися за допомогою стандартних механізмів .NET.

Цілісність і несуперечливість даних при роботі з обома платформами підтримується за допомогою механізму розподілених транзакцій.

Відмовостійкість додатків в .NET, як і в EJB, повинна забезпечуватися або за рахунок використання додаткових продуктів, або за рахунок специфічного проектування програми.

Захищеність додатків для обох платформ підтримується такими методами: техніки аутентифікації, можливість визначення ролей, забезпечених набором прав доступу до різних елементів системи, а також можливість використання різних протоколів шифрування і захищеної передачі даних, управління ключами і підтвердження цілісності даних.

У EJB робота з XML здійснюється за допомогою спеціальної бібліотеки, що стосується .NET, то можливість роботи з XML-даними є вбудованою в рамках механізмів ADO.NET [4].

Обидві технології, EJB та .NET, можуть використовуватися для розробки розподілених компонентів, але їх також можна використовувати локально. Це, однак, більш вірно для компонентів .NET, ніж для компонентів EJB, оскільки компоненти EJB оброблюються контейнером, клієнт якого підключається віддалено, якщо тільки клієнт не є іншим компонентом або сервлетом, розміщеним на одному і тому ж контейнері або сервері додатків відповідно.

.NET має сильні інструменти для інтерфейсу користувача і краще розвинену клієнтську сторону. В той же час Java має перевагу на боці сервера, оскільки сам по собі JEE являє собою набір специфікацій, сфокусованих на полегшення розробки серверних додатків, зокрема на Unix-системах[4].

Однією з сильних сторін EJB є те, що широкий вибір інструментів, продуктів і додатків доступний від багатьох постачальників. Конкуренція серед них сприяє постійному вдосконаленню технології. Розвиток же .NET повністю залежить від Microsoft, однак ця технологія молодша на декілька років, і при її створенні були враховані деякі недоліки вже існуючих технологій.

Технології EJB і Microsoft .NET аналогічні за своєю природою, кожна з них має певні переваги і недоліки. Обидві технології в багатьох аспектах мають свій власний спосіб і тип реалізації для вирішення різних завдань, наприклад, для віддалених викликів. Крім цього, у них є підтримка багатьох подібних функцій.

Обираючи між двома технологіями, в першу чергу потрібно зважати на те, де буде використовуватись розроблювана система, які вимоги до неї висувуються, що є більш важливим: багатоплатформенність чи можливість вибору мови програмування, більш сильна клієнтська сторона чи серверна, участь у створенні системи розробників, які досконало знають одну з технологій, або вже наявне програмне забезпечення, що буде взаємодіяти з системою або використовуватись для розробки, також може стати одним з критеріїв вибору.

Література:

1. Смертельная схватка: .NET против J2EE [Электронный ресурс] – Режим доступа: <http://www.javaportal.ru/java/articles/netvsj2ee.html>. Дата доступа: 25.04.17.
2. Java vs .NET: почему .NET [Электронный ресурс] – Режим доступа: <http://www.shapovalov.org/news/2010-05-12-678>. Дата доступа: 29.04.17.
3. J2EE vs. Microsoft.NET [Электронный ресурс] – Режим доступа: http://www-01.ibm.com/software/smb/na/J2EE_vs_NET_History_and_Comparison.pdf. Дата доступа: 3.05.17.
4. Кулямин В. В. Технологии программирования. Компонентный подход / В. В. Кулямин. - Москва: ИСП РАН, 2006. – 315 с.