ЮБИЛЕЙНА НАУЧНА СЕСИЯ – 30 години ФМИ
ПУ “Паисий Хилендарски”, Пловдив, 3-4.11.2000

МНОГОСЛОЙНА АРХИТЕКТУРА ЗА БИЗНЕС-ПРИЛОЖЕНИЯ

Емил Николов Хаджиколев

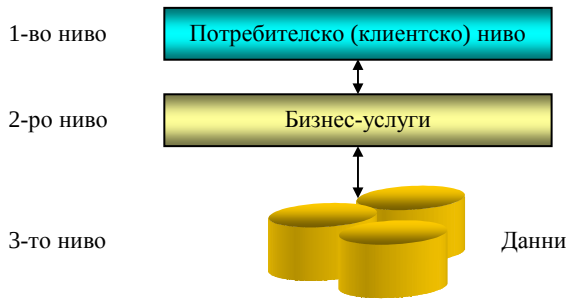
В статията са разгледани някои основни характеристики на еднослойни, двуслойни, трислойни и многослойни архитектури на бизнес-приложения. Обърнато е по-голямо внимание на технологията Enterprise Java Beans за изграждане на средния слой (сървър за приложения) в една модерна архитектура за бизнес-приложения. Във втората част на публикацията е представен проекта с работно заглавие MLABA (MultiLayer Architecture for Business Applications – Многослойна архитектура за бизнес-приложения) за изграждане на бизнес-компоненти на сървъра за приложения. Освен това се обсъждат първите резултати. Проектът е съставна част на проекта M-STEB [1].

Ключови думи: *многослойна архитектура, сървър за приложения, бизнес-навигатор*

Въведение в многослойните архитектури

Изграждането на едно бизнес-приложение изисква използването на множество софтуерни услуги. За работа с потребителя е необходимо създаването на подходящи входни екрани за въвеждане на информация и съответно – екрани за извеждане на получени резултати. Друг тип услуги са алгоритмите, прилагащи бизнес-правилата за съответния бизнес, за който е създадено приложението. Бизнес-услугите обработват входните данни, получени от потребителя, както и данни, съхранени в бази данни (или файлове). Те осъществяват логиката на работа на приложението. Третият вид услуги са свързани със записването на част от данните, получени в резултат на работата на бизнес-системата, в базата данни и четенето им.

Това логическо разделяне на функциите на бизнес-приложението на отделни нива е илюстрирано на Фиг.1. На клиентското ниво се осъществява комуникацията между човека и приложението. В него от една страна се обработва информацията получена от потребителя (въведена чрез мишка, клавиатура или друго входно устройство) и се предоставя за по-нататъшна обработка, а от друга – по подходящ начин на екрана се извежда създадената от приложението информация. Най-важното



Фиг 1. Логическа архитектура на бизнес-приложение

ниво е второто – освен че предоставя бизнес-логиката на приложението, в него се намират функции за обработка на входно-изходната информация както за клиента, така и за данните от третото ниво.

Реално броят на нивата в едно приложение се определя в зависимост от физическото разделяне на функциите в логически ясно обособени отделни модули, които могат да се използват многократно и от други приложения. Възможно е функциите да се разделят в повече от 3 нива.

Еднослойна архитектура

При еднослойните приложения логическите нива са тясно свързани. Функциите от всяко ниво познават ясно логиката на другите нива. Използването на тази архитектура е подходящо при създаването на еднопотребителски приложения, които работят на един компютър. Недостатък на тази архитектура е, че при промяна на бизнес-правилата цялата структура на приложението трябва да бъде преправена и съответно всеки потребител трябва да си инсталира новата версия. Освен това, възниква трудност при нуждата в една компания да се използват общи данни от няколко различни приложения.

Двуслойна архитектура

При двуслойната архитектура нивото на данните е отделено в самостоятелен модул, наречен система за управление на бази данни. Потребителското и нивото на бизнес-логиката остават тясно свързани. Тук, както и при архитектурите с повече нива, имаме архитектура от тип клиент-сървър (server – слуга), при която клиента иска услуга от сървъра. Клиент-сървър приложенията обикновено работят в мрежова среда, като множество клиенти могат да имат достъп до един сървър едновременно. Приложението клиент и приложението сървър могат да съжителстват и на един компютър.

Достъпът на клиентското приложение до сървъра за бази данни става посредством стандартен език, който трябва да бъде поддържан от сървъра. За работа с най-използваните в момента бази данни – релационните – се използва езикът SQL (Structured Query Language).

Трислойна архитектура

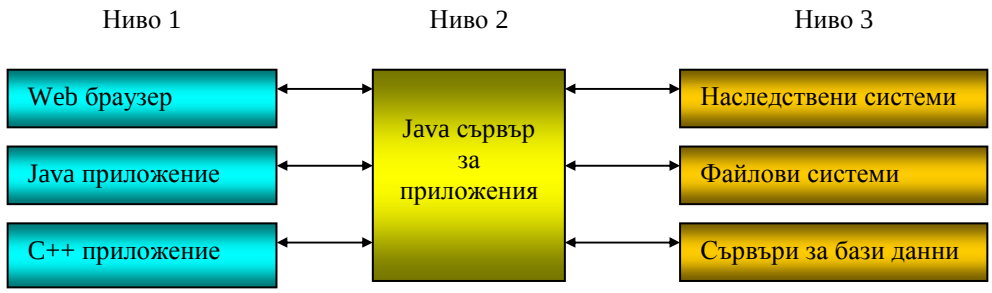
Благодарение на бързото развитие на Интернет и софтуерните технологии все повече потребители използват едни и същи услуги за достъп до информация, свързана с електронната търговия. Използването на стандартизирания потребителски интерфейс (Web браузер) на Интернет намалява значително разходите за достъп до информацията. Интернет се превръща в сцена за представяне, предлагане и купуване на продукцията на големи и малки търговски фирми. Главните вариации при използването на Web браузер за достъп до данни са:

- двуслойни архитектури, използващи Java аплети като клиенти, поддържащи бизнес-логиката
- трислойни архитектури, в които средното ниво е Web сървър, предлагащ разнообразни бизнес-услуги. Услугите са програми, написани на език за програмиране като Java, C++, Perl…

При трислойната архитектура клиентското приложение не се влияе от бизнес-логиката или от базата данни. Възможно е да се промени бизнес-логиката или стратегията за съхраняване на данните без това да повлияе на начина на представяне на информацията при клиента.

Многослойна архитектура

Вместо Web сървър като среден слой в една трислойна архитектура може да се използва сървър за приложения (Application Server). Той позволява голям брой програми да се изпълняват и да обменят информация помежду си, като използват различни протоколи за комуникация. По такъв начин множество мощни приложения, работещи на различни платформи и написани на различни езици за програмиране, предоставят различни услуги за различни клиенти [6]. Изграждането на разпределени приложни сървъри е нов етап от развитието на мрежовите приложения. Приложението сървър работи на няколко различни компютъра. Специална услуга следи за натоварването на системните ресурси и определя къде да се изпълни заявката. Поради горните особености се счита, че архитектура от този тип е многослойна. Недостатъци на сървърите за приложения са високите цени (десетки хиляди долари) и високите изисквания към системните ресурси.



Фиг. 2. EJB клиент-сървър трислойна (многослойна) архитектура

Модел на многослойна архитектура, изграден въз основа на Enterprise Java Beans (EJB) технологията е представен на Фиг.2. Тази технология е част от платформата Java 2 Enterprise Edition (J2EE), предоставяща архитектура за развитие, разгръщане и изпълнение на приложения в разпределена среда. Приложенията създадени върху тази платформа използват множество стандартни системни услуги като нишки, поделяне на ресурсите, управление на транзакциите, сигурност, връзка с клиента и достъп до бази данни. Като услуга на сървъра е предвидено и реализирането на Web контейнер, изпълняващ ролята на Web сървър. Предоставяйки тези услуги, EJB платформата позволява на разработчика да се фокусира върху бизнес-логиката на приложението. Кодирането на логиката става в enterprise бийнове (enterprise beans) – компоненти за многократна употреба, които могат да бъдат достъпни от клиентската програма. Enterprise бийновете се изпълняват на сървъра за приложения. Той може да поддържа множество от клиенти като Web браузери, Java приложения, приложения написани на езици за програмиране, поддържащи протокола за заявки към обекти CORBA (Common Object Request Broker Architecture). Клиентите работят с потребителския интерфейс. Прозрачно за тях сървърът за приложения се обръща към базите данни и наследствените системи, изпълнява бизнес-правилата върху данните и връща резултат [5].

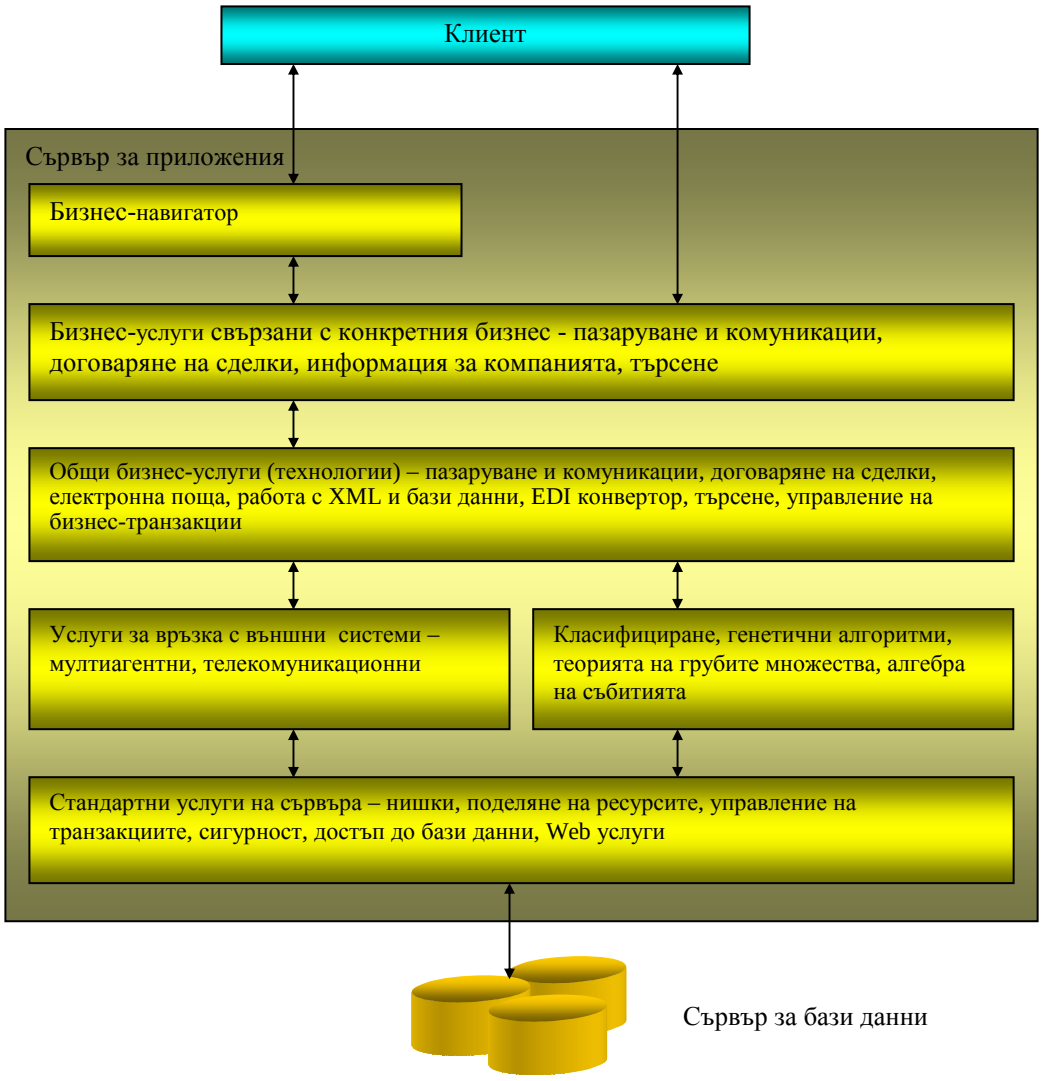
Използването на трислойни и многослойни архитектури вместо двуслойни може значително да намали общия брой на връзките между клиентите и базите данни. Нека например, имаме N клиента и M сървъра за бази данни като всеки клиент желае достъп

до всеки от сървърите. При двуслойна архитектура, връзките между тях са N*M, докато в трислойните архитектури общият брой на връзките между клиентите, приложния сървър и сървърите за бази данни може да се сведе до N+M [7].

Проект MLABA

Сървърът за приложения предлага среда за изпълнение на сигурни бизнес-услуги. Добавянето на стандартни бизнес-технологии и услуги може да допринесе за създаването на един доста атрактивен сървър за бизнес-приложения.

Изграждането на такива бизнес-компоненти е приоритет на проекта MLABA, който е в процес на реализация. Те ще предоставят средства за лесното изграждане на услуги,



Фиг. 3. Услуги на сървъра за бизнес-приложения

свързани с конкретен бизнес и управление на сложни бизнес-процеси. Проектирано е създаването на множество компоненти, по-важни от които са: бизнес-навигатор и свързаният с нея компонент за управление на бизнес-транзакции; EDI компоненти, доставящи методи за обработка и съхранение на документи за електронна търговия; компоненти за търсене в подредени (XML) и неподредени документи; компоненти, използващи алгоритми за класифициране, генетични алгоритми и теорията на грубите множества. Създаването на компоненти за връзка към разпределена мулти-агентна среда за изпълнението на част от задачите на сървъра, както и включването на телекомуникационни услуги е друга важна задача на проекта, която е свързана и с проекта M-STEB.

На Фиг.3 е показана вътрешната архитектура на сървър за бизнес-приложения.

На най-ниско ниво са стандартните услуги на сървъра за приложения – нишки, поделяне на ресурсите, управление на транзакциите, сигурност, връзка с клиента и достъп до бази данни.

Използването на мулти-агентната среда, която реализира част от бизнес-услугите би могло да повиши скоростта и ефективността на изпълнението им. Вграждането на интерфейси към телекомуникационни услуги би увеличило сферата на влияние на сървъра, с възможност за достъп до бизнес-услугите от мобилни телефони и други телекомуникационни апарати.

Теорията на грубите множества [8], алгоритмите за класифициране [9] и генетичните алгоритми [11] са средства за решаване на сложни проблеми за търсене.

Създаването на EDI компоненти и компоненти за търсене в бизнес-документи е свързано с използването на езика XML. EDI (Electronic Data Interchange) е стандарт за обмен на данни по електронен път, който се използва в световен мащаб за бизнес-към-бизнес (B2B) комуникации. Езикът XML за работа в Web се появава не само като алтернатива и разширение на HTML (едни XML данни могат да бъдат представени в различни форми), но той си поставя целта да стане универсален език за обмен на данни между приложения. XML разделя структурата, съдържанието и представянето на данните. XML документите са разбираеми както от потребителя, така и от компютъра. В тях може да се провежда по-смислено търсене, основано на семантиката на документите [4]. За да бъде представен бизнеса им в Интернет, все повече фирми преминават към използването на XML за обмен на бизнес-документи. Осигуряването на интерфейс към наследствените EDI системи може да се реализира чрез EDI/XML конвертор. Създадени са Java класове (от които лесно могат да бъдат получени Java компоненти) на един универсален EDI/XML конвертор [2], който сме тествали за конкретен вид бизнес-документи (ETO - Electronic Trade Opportunity – подстандарт на EDI). По отношение на търсенето в неструктурирани текстови документи също имаме конкретни резултати – предоставя се възможност за търсене на логически израз от думи (с използването на логически съюзи и, или, не, както и на скоби) в документ, като самите думи могат да бъдат непълно зададени (по определени правила) [3].

Компонентът “бизнес-навигатор” служи за създаването и управлението на работни потоци (workflow). Отговаря за получаването на всички заявки от страна на клиента, определя вида и смисъла на заявката, разделя я на отделни подзадачи, водещи до решението й, и ги разпределя на нужните за изпълнението им компоненти. Изпълнението на подзадачите се следи от компонент за управление на бизнес-транзакции. Подходящ входен език за заявки към бизнес-навигатора е XML.

Друг вариант за внасяне на интелигентност в приложния сървър е прилагането на алгебрата на събитията [10]. Тя изисква обединяване на компонентите в обща система за получаване и предаване на събития. Всеки компонент се задейства в зависимост от настъпването на определени събития в системата. При работата си той създава нови събития, които задействат нови компоненти. Изпълнението на една заявка започва след генериране на събитие за пристигнала заявка. Бизнес-навигаторът създава събитие за типа на заявката, което се прихваща от компонент, обработващ този тип заявки, като той може да задейства и други компоненти. Накрая се генерира събитие за завършено изпълнение на задачата и бизнес-навигатора предава резултата на клиента.

Заключение

Представената в публикацията многослойна архитектура използва един стандартен EJB модел. Добавянето на предложените от нас бизнес-компоненти на сървъра за приложения доставя универсална и мощна основа за изграждане на различни бизнес-приложения. Наличието на интелигентен компонент за динамично генериране и управление на бизнес-логика (бизнес-навигатор) осигурява успешното и бързо изпълнение на сложни заявки, което е критичен фактор за електронния бизнес в Интернет.

ЛИТЕРАТУРА

[1] С. Стоянов; И. Ганчев, “Мулти-агентна технология за изграждане на информационни системи за електронна търговия”, Юбилейна научна сесия 30 години ФМИ, Пловдив (прието за печат).

[2] С. Ламбова, “Използване на XML и EDI за електронна търговия. XML/EDI framework” (Дипломна работа), ЙУ “П. Хилендарски”, 2000.

[3] Н. Атанасов, “Интегрирана система за търсене в текстови документи” (Дипломна работа), ЙУ “П. Хилендарски”, 2000.

[4] D. Chang, D. Harkey, “Client/Server Data Access with Java and XML”, New York, John Wiley & Sons Inc., 1998.

[5] Overview of Enterprise Java Beans technology, http://web2.java.sun.com/j2ee/j2sdkee/techdocs/guides/ejb/html/Overview.fm.html

[6] “One, two, three, or n tiers?”, http://www.javaworld.com/javaworld/jw-01-2000/jw-01-ssj-tiers_p.html

[7] A White Paper about N-tier architectures, http://www.cis.temple.edu/~ingargio/cis307/readings/corba.html

[8] Z. Pawlak, “Rough Sets. Theoretical Aspects of Reasoning about Data”, Institute of Computer Science, Warsaw, 1990.

[9] M. Lenz, B. Rartsch-Spoerl, H. D. Burkhard, S. Wess, “Case-Based Reasoning Technology”, From Foundations to Applications, Springer-Verlag, 1998.

[10] M. N. Huhns, M. P. Singh, “Workflow agents”, IEEE Internet Computing, Vol.4 (1998), pp.94-96.

[11] Koza, R. John, “Genetic Programming: On the Programming of Computers by Means of Natural Selection”, Cambridge, MA: The MIT Press, 1992.

ул. “Цар Иван Асен II” 110, гр. Асеновград,
тел. 0331 24517, e-mail:

MULTILAYER ARCHITECTURE FOR BUSINESS APPLICATIONS

Emil Nikolov Hadjikolev

In this paper I give some basic characteristics of 1-tier, 2-tier, 3-tier and n-tier architectures for Business Applications. I take notice of Enterprise Java Beans technology, which I use to build the middle tier (Application server) of one modern architecture for Business Applications. In the second part of this paper I represent a project named MLABA (MultiLayer Architecture for Business Applications), which purpose is to create some buisiness components for the Application server. In addition I discuss the first results. This project is a part of the project M-STEB [1].