WEB 2.0 АРХИТЕКТУРА НА DeTC НА БАЗАТА НА MORFIK И JST

Доц. д-р Асен Рахнев, ФМИ, ПУ „Паисий Хилендарски", Пловдив
Гл. ас. д-р Олга Рахнева, УХТ, Пловдив
Стойчо Монев, ФМИ, ПУ „Паисий Хилендарски", Пловдив

WEB 2.0 ARCHITECTURE OF DeTC BASED ON MORFIK AND JST

Assoc. Prof. Asen Rahnev, PhD, FMI, Plovdiv University, Plovdiv
Ch. Ass. Olga Rahneva, PhD, UFT, Plovdiv
Stoicho Monev, FMI, Plovdiv University, Plovdiv

Abstract

This paper describes the modern technologies Web 2.0, Morfik and JST. They are used to implement a new Web 2.0-compliant architecture for the Distributed e-Testing Cluster – DeTC (Distributed e-Testing Cluster).

1. Въведение

Разпределеният клъстер за електронно тестване – DeTC (Distributed e-Testing Gluster) е един нов подход за мрежово базирано електронно тестване, където взаимодействат физически разпределени интерактивни и кооперативни единици, помощни средства, обучаеми, обучаващи и администратори [6]. Бързопроменящият се свят на уеб технолгиите предоставя отлична възможност за разширяване на възможностите на DeTC чрез използване на най-новите разработки в областта, позволяващи създаването на високо интерактвини уеб приложения. От друга страна развитие на уеб технолгиите води до създаването на нови, по-мощни инструменти за разработка на приложения, позволяващи да се създават комплексни уеб приложения, с високо ниво на интерактивност.

В тази работа се описват новите технологии Web 2.0, Morfik и JST. На тяхна база се предлага нова Web 2.0 архитектура на разпределения клъстер за електронно тестване – DeTC.

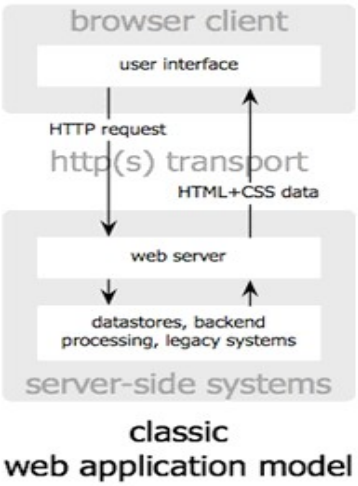
2. Web 2.0

Основен проблем със съществуващите технологии и техники е че връзката между човек, група и обществото изисква три изцяло различни парадигми: настолна операционна система, мрежова операционна система и световната мрежа (Интернет). Нуждаем се от съвършено нов подход към настолната операционна система и в частност - операционните системи в мрежата. Операционни системи, които ще са обърнати към потребителя, а не потребители, които да се съобразяват с операционните системи.

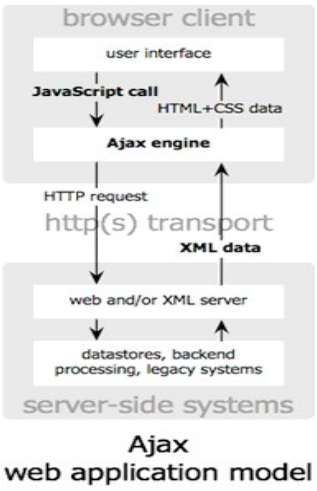
В основата на Web 2.0 [4] стои Ajax (Asynchronous JavaScript and XML). Ajax [3, 5] е похват в уеб разработките за създаване на интерактивни уеб приложения. Идеята е уеб страниците да станат по-интерактивни чрез асинхронен обмен на малки порции данни „зад кадър“, така че да не бъде необходимо да се презарежда цялата страница отново.

За разлика от класическият уеб, чрез Web 2.0 се повишава интерактивността, скоростта и функционалността на страниците. Намалява се натовареността на уеб сървърите, потребителите по-бързо си свършват работата, защото приложенията са по-бързи, по-интуитивни и достъпни, и в крайна сметка се повишава ползваемосттта на уеб приложенията.

В класичската уеб архитектура (Фиг. 1) всяка заявка от киента към сървъра предизвиква обновяване на цялата страница, което означава генериране на страницата на сървъра, изпращането на страницата до клиента и обработката на страницата от браузъра. При Ajax архитектурата (Фиг. 2) заявката към сървъра не води до обновяване на цялата страницата, а до частична промяна.

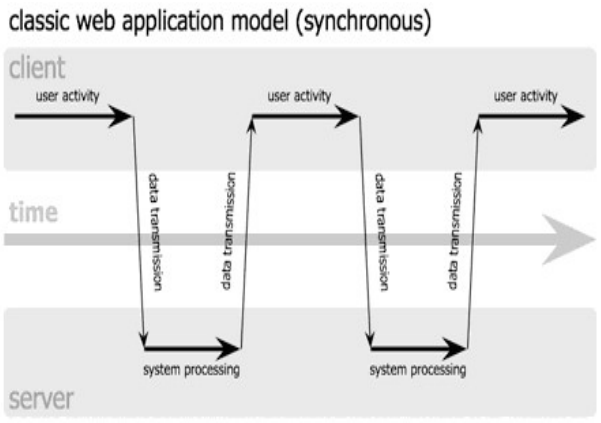


Фиг. 1 Класическа Web архитектура

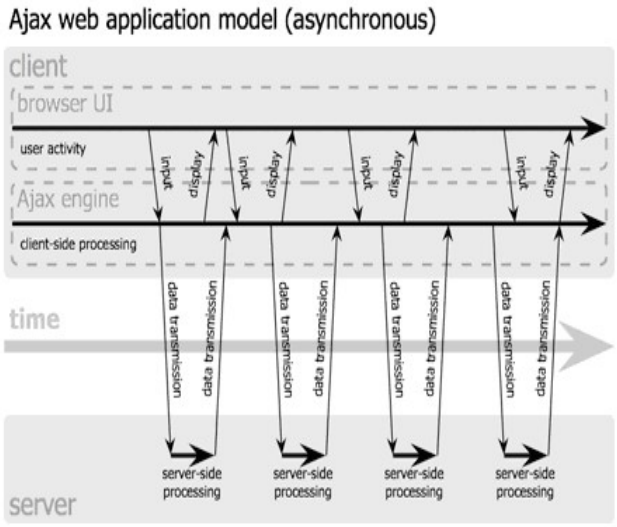


Фиг. 2 Ajax Web архитектура

Синхронното (Фиг. 3) и асинхронното изпълнение на заявки са една от основните разлики между класическият уеб и Web 2.0. При асинхронното изпълнение на заявките (Фиг. 4), потребителят може да изпрати заявка и да продължи работата си, без прекъсване и чакане на отговор от сървъра. Когато сървърът е готов с обработката на заявката, той връща резултата и с подходящ потребителски интерфейс браузърът съобщава на потребителя за това.



Фиг. 3 Синхронно изпълнение на заявките в класическа Web архитектура



Фиг. 4 Асинхронно изпълнение на заявките в Web 2.0 архитектура

3. Morfik и JST

Morfik [2] е среда за бърза разработка на Web 2.0 приложения с разработена автоматизация на целия този Ajax цикъл, произведена от едноименната компания [1]. Операционната система, в която работи приложение, направено с Morfik е браузърът. Разработчикът няма нужда да се грижи за поддръжка на най-различни браузъри, което е проблем, отнемащ сериозен процент от времето за разработка на Уеб приложение. В момента се подържат новите версии на Firefox, Internet Explorer, Opera и Safari. Приложенията, написани с Morfik не изискват познания по Javascript, HTML или CSS – основните технологии зад всяка уеб страница.

Ядрото ма Morfik е JST - JavaScript Synthesis Technology. Езиците, на които се пише код в Morfik са няколко: Object Pascal, C#, Java и Visual Basic, като

тук имаме предвид синтаксиса на езиците. Всеки модул се компилира до JST и от там се генерира:

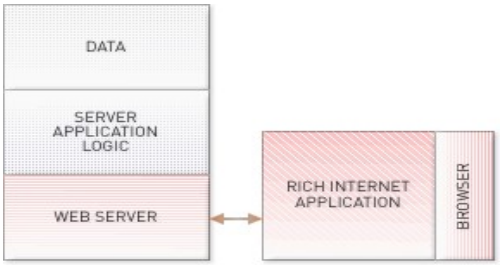
- клиентски код (HTML, JavaScript и CSS)
- сървърен код (exe за Windows, so за Linux и съответният файл за Mac OS)
- интеграция с базата данни (Firebird по подразбиране, може да си използват и други бази данни)

Има възможност за интеграция и ползване на функционалност предоставена от външни Javascript библиотеки, както и на външни динамични библиотеки за Windows (dll).

Morfik предоставя възможност за използване и предоставяне на уеб услуги. Уеб услугите са съвременна Интернет-базирана технология, която представлява своеобразно развитие на предшестващите я технологии за разработка на разпределени приложения. Технологията представлява стандарт за използване на приложни компоненти през различни отворени протоколи, като компонентите могат да взаимодействат помежду си. Стандартизацията обхваща комуникационните протоколи, представянето на данни – в XML формат, езиците за описание на компонентите и механизмите за откриване. Комуникацията се постига чрез специален протокол – SOAP, базиран на XML, при който съобщенията към и от компонентите се форматират в XML.

В Morfik уеб услугите се използват чрез дефиниране на методи на сървъра и изискването им от клиента чрез интуитивна и не толкова сложна форма, каквато е комбинацията от SOAP и XML.

4. Web 2.0 архитектура на DeTC



Фиг 5. Четирислойна архитектура на DeTC

Четирите слоя, представляващи архитектурата на приложение на DeTC са показани на фигура 5: база данни, сървър за приложения, уеб сървър, браузър.

В базите данни се съхраняват данните на приложението, данните за потребителите на приложението, и някои системни данни. Като например, всяка таблица има полета, попълвани автоматично за последни редакции на съществуващи записи или за нови записи. Тези полета в последствие се използват при задачи за синхронизация на данните в различни бази.

Сървърът за приложения е междинният слой между базата и уеб сървъра. Той предоставя набор от уеб услуги на клиентите, като:

- достъп до данните
- модифициране на данните

- услуги за сигурност и права на потребителите

Сървърът за приложения е многонишков, като всеки клиент се обслужва от отделна нишка.

Уеб сървърът представлява междинен слой на комуникацията между клиентските приложения и сървъра за приложение. Той няма достъп до базата данни. Освен това, уеб сървърът се грижи за коректното разпределение на HTTP заявките, и предоставя възможност за инсталиране на сертификати за сигурност на комуникациите. В Web 2.0 версията на DeTC използваме уеб сървъра Apache, като предстои интергация и с Internet Information Services (IIS) – уеб сървърът на Майкрософт.

Клиентските приложения се изпълняват в браузър, който предоставя общ потребителски интерфейс, манипулиране на форми, обмяна на данни между формите. Те работят на принципа на тънкия клиент, където всички данни се намират на сървъра и се извикват само когато е необходимо. Клиентската част на приложението на Morfik дава възможност да се създават богати уеб приложения.

Всяка уеб форма представлява таблица от базата или SQL заявка от базата и уеб компоненти за манипулирането им, генерирани автоматично от средата за разработка. Изгледът на формата съдържа разбивка на таблицата на части, за да се предпази уеб сървъра от излишно натоварване и извикването на всички данни, когато това не е нужно, както и за прегледност на потребителския интерфейс. Например, в повечето случаи изпитваните нямат нужда от едновременен преглед на всички въпроси от даден тест.

Една типична заявка от страна на клиента представлява HTTP заявка, която браузърът изпраща до Уеб сървъра, който я препредава на сървъра за приложение, където в зависимост от правата, получени от клиента, тя се обработва, ако трябва се прави заявка до базата данни или се взема кеширан обект от паметта и от там заявката се връща обратно към клиента.

Важно е да се отбележи, че по принцип в света на уеб приложенията всички грижи за сигурността и правата за използване на дадени обекти от базата данни, трябва да са най-стриктни на ниво сървър за приложение. Не можем да разчитаме на сигурност в клиентското приложение, защото данните там са лесни за манипулиране. В приложението използваме сесии за защита на данните.

Данните на сървъра могат да се кешират автоматично. Това позволява когато се изискват данните от дадена страница, те да не трябва да се извличат всеки път чрез заявка от базата, когато това не е нужно, а направо от паметта, което е много по-бързо. Този проблем е решен като е оставена свобода на разработчика – дали да поиска обновяване на данните, или да просто да вземе кешираните данни. Предоставена е възможност на крайния потребител да прецени и ако му е нужно изрично да изиска от сървъра обновени данни.

Web2.0 базираната версия на DeTC представлява приложение, разработено използвайки Morfik, позволяващо провеждане на изпитни тестове отдалечено. Има възможност и за работата му през локална мрежа. И в двата случая приложението работи в браузър, което спестява доста усилия за разпространяване на програми на всички компютри, на които ще се провежда изпита. Изпитващият може да дефинира тестове, различни видове въпроси, да генерира шаблонни въпроси (един и същ въпрос, с различни входни параметри),

може да дефинира времетраене, може да дефинират диапазони за автоматично оценяване на изпитните тестове.

При разработката е обърнато внимание на времетраенето на провеждането на тестове, като това време се управлява от сървъра, а не от клиентският компютър. Така няма възможност за удължаване на времето за изпитване от страна на самият изпитван чрез промяна на системните настройки (времето) на компютъра, на който се провежда изпита. Направена е интеграция с настолната версия на приложението за изпитване, така че данните за изпити, тестове, въпроси, отговори могат да се репликират двупосочно, за използване на данните и на двете места.

Избрали сме приложението да бъде с Unicode (utf-8) кодировка. За база данни сме избрали Firebird също с Unicode кодировка (Unicode_FSS). Проблемът с интернационални версии на настолна приложения е лесно решим в Web 2.0 версията на DeTC.

Предвидена е възможност всеки тест може да бъде експортиран към pdf , за самостоятелна подготовка или за преглед на изпитни резултати.

5. Благодарности

Този проект е частично финансиран от Фонд „Научни изследвания“ към МОН по договор ВУ-МИ 107 / 2005

Литература

[1] http://morfik.com

[2] THE WEB OPERATING SYSTEM
http://morfik.com/WebOS_Discussion_Paper.pdf

[3] Ajax: A New Approach to Web Applications
http://www.adaptivepath.com/ideas/essays/archives/000385.php

[4] Web 2.0, http://en.wikipedia.org/wiki/Web_2

[5] Ajax, http://en.wikipedia.org/wiki/Ajax_%28programming%29

[6] A. Rahnev, O. Rahneva. N. Pavlov, “Architecture & Design of Distributed Electronic Testing Cluster (DeTC) based on Microsoft .NET Framework” – IMAPS CS International Conference 2005, September 15-16, 2005, Brno, Czech Republic, pp 417-422.