

PHP CLOSER TO DESKTOP

Aleš Rybák

Master Degree Programme (6), FIT BUT

E-mail: xrybak00@stud.fit.vutbr.cz

Supervised by: Aleš Smrčka

E-mail: smrcka@fit.vutbr.cz

ABSTRACT

The project is aimed on simplification and efficiency increasing of programmer's work while developing web applications. The main goal is to provide the PHP programmer with an object framework which makes web application development similar to the development of desktop applications. This is accomplished by automatic saving of application state by the framework and its modification using callback functions.

1 ÚVOD

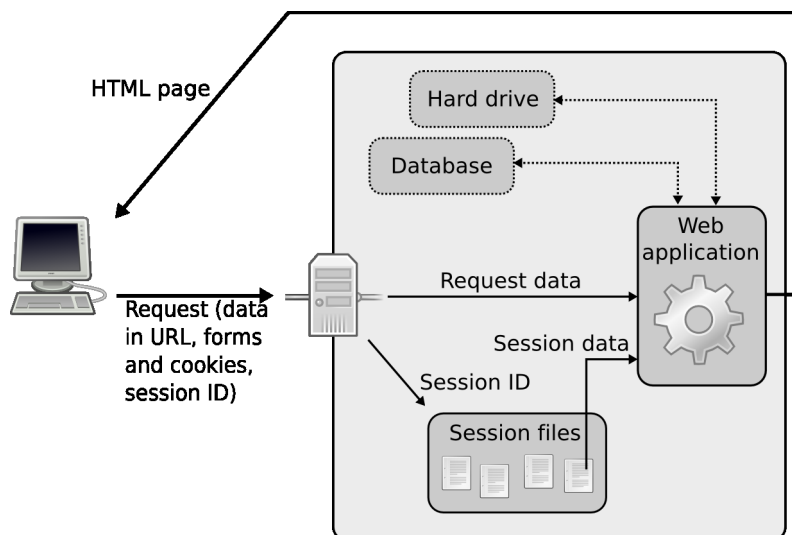
Zvyšující se výkon výpočetní techniky umožnil během posledních let také rozvoj internetových technologií. Na původně zcela statických stránkách se začaly objevovat dynamické prvky a zanedlouho se objevily stránky generované zcela dynamicky, které poskytují funkčnost podobnou desktopovým aplikacím. Projekt PHP Closer to Desktop se zaměřuje na usnadnění a zvýšení efektivity práce programátora, který chce vyvíjet právě takové aplikace pomocí skriptovacího jazyka PHP.

2 SOUČASNÁ SITUACE

Na Internetu a zejména pak v intranetových sítích se stále častěji objevují webové aplikace, které poskytují rozmanité služby velmi často se blížíci službám dosud nabízeným desktopovými aplikacemi. Tento trend je patrný i nástrojů jako jsou javové applety, prvky ActiveX či AJAX (viz [3]). Nevýhodou těchto nástrojů oproti čistě webové aplikaci je, že nutí uživatele k instalaci software na jeho počítač, omezují se pouze na konkrétní operační systém nebo vyžadují konkrétní nastavení prohlížeče. Webové aplikace těží právě z dobré dostupnosti i pro laické uživatele, protože nevyžadují instalaci žádného software – vystačí pouze s internetovým prohlížečem, který je v současnosti nedílnou součástí snad všech operačních systémů, a nejsou tak závislé na konkrétní platformě.

Rozdíl mezi desktopovými a webovými aplikacemi z pohledu programátora je zejména v přístupu k datům aplikace. Klasická desktopová aplikace má svá data umístěna v paměti uživateleova počítače. Tato data reprezentují stav dané aplikace a ta k nim má přístup po celou dobu běhu aplikace. Když je aplikace ukončena, jsou data z paměti odstaněna. Naproti tomu webová

aplikace jako taková je bezstavová. Uživatel zašle serveru požadavek a server spustí skript, kterému předá vstupní parametry, na jejichž základě je vygenerována stránka. Stav webové aplikace v paměti serveru je po vygenerování stránky odstraněn. Proto je nutné stav webové aplikace uchovávat jiným způsobem. Těchto způsobů je více (viz obrázek 1), z nichž každý má své výhody a nevýhody v rychlosti zpracování, objemu dat přenášeném mezi serverem a klientem, bezpečnosti, aj. Jednu vlastnost mají ovšem společnou, a to že programátor musí stav aplikace uchovávat ručně a v mnoha případech i složitě.



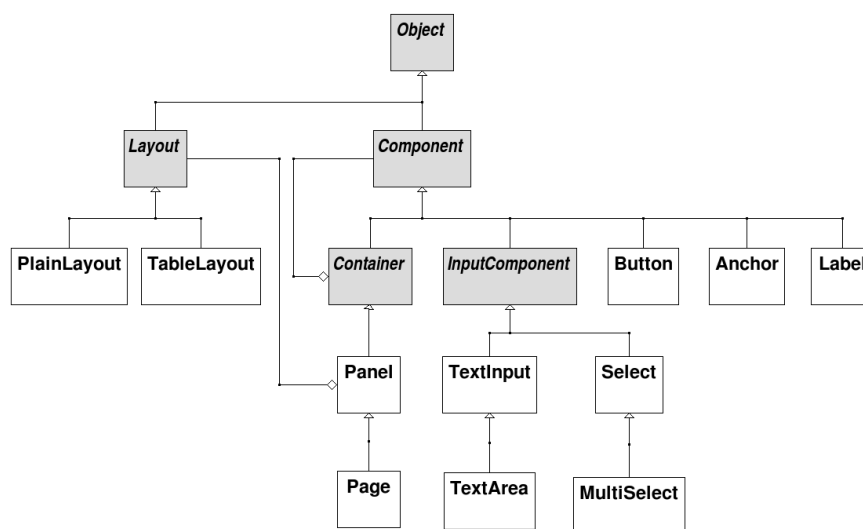
Obrázek 1: Zpracování požadavku webovou aplikací

3 FRAMEWORK

Cílem projektu PHP Closer to Desktop tedy je poskytnout framework, jež automaticky zajišťuje uchování stavu aplikace tak, aby se o něj programátor nemusel starat explicitně.

Počítáme-li s tím, že máme zajištěno uchování stavu aplikace, budeme tento stav chtít nějakým způsobem modifikovat. V desktopových aplikacích je stav měněn prostřednictvím modifikačních funkcí (tzv. zpětná volání, angl. callback functions) a nejinak tomu je i našeho frameworku. Princip modifikačních funkcí, které obsluhuje framework nabízí možnost vytvářet aplikace podle návrhového modelu model–view–controller (zkráceně MVC, viz [4]), jenž je v současnosti velmi často používán zejména proto, že rozděluje celou aplikaci do několika vrstev, které jsou mnohem přehlednější, lépe spravovatelné a odolnější proti chybám.

Framework je postaven do role řadiče. Datová vrstva (model) je zajištěna objekty, jejichž stav je uchovávan pro daného uživatele pomocí frameworku. Vývoj prezentační vrstvy umožňuje řada komponent, které namísto přímého vytváření HTML kódu poskytují objektově orientovaný přístup. Programátor pracuje čistě s objekty v jazyce PHP, které reprezentují jednotlivé entity, a ty potom generují vlastní HTML kód. Pomocí dědění z již vytvořených komponent (viz obrázek 2), lze vytvářet znovupoužitelné entity s rozšířenou funkcionalitou (např. vstupní pole formuláře akceptující pouze čísla).



Obrázek 2: Vybrané třídy používané pro vytváření prezentační vrstvy webové aplikace

4 ZÁVĚR

Nejvýznamějším přínosem projektu je automatické ukládání stavu bez zásahu programátora, které zmenšuje propast mezi desktopovými a webovými aplikacemi. Objektový přístup a změna stavu aplikace pomocí callbackových funkcí navíc dovoluje implementovat aplikaci podle návrhového vzoru MVC a oddělit tak prezentační a datovou vrstvu. Implementace datové vrstvy je podporována automatickým ukládáním stavu, prezentační vrstva pak množstvím objektů, které generují validní HTML kód.

Projekt tedy umožňuje programátorovi efektivnější přístup k vývoji webových aplikací, zjednodušuje jejich správu či případnou pozdější modifikaci a je koncipován tak, aby mohl být dále rozvíjen a rošiřován. Navíc je otevřena možnost tvorby prezentační vrstvy pomocí grafického návrhového studia, tj. aplikace umožňující práci s objekty prostřednictvím grafického rozhraní.

REFERENCE

- [1] Oprštný, M.: PHP Modular Object framework [diplomová práce]. Vysoké učení technické v Brně, Fakulta informačních technologií, 2006.
- [2] Rybák, A.: Balíček modulů pro tvorbu webových aplikací pomocí PHP [semestrální práce]. Vysoké učení technické v Brně, Fakulta informačních technologií, 2006.
- [3] Darie, C., Brinzarea, B., Chereches-Tosa, F., Bucica, M.: AJAX a PHP - tvoříme interaktivní webové aplikace profesionálně. Zoner Press 2006.
- [4] Baray, C.: The Model–View–Controller Design Pattern. Indiana University 1999. URL: <http://www.cs.indiana.edu/~cbaray/projects/mvc.html>