
Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach

Žilina 2.–5. júla 2009



WEB2PY FRAMEWORK NA TVORBU WEBOVSKÝCH APLIKÁCIÍ

KARABÁŠ, Ján (SK), SILÁČI, Jozef (SK), ŠUCH, Ondrej, (SK)

1 Úvod

Programovanie webovských stránok je najrozšírenejším druhom programovania. Zamestnávateľia očakávajú od absolventov skúsenosti v najnovších, najproduktívnejších webovských technológiach. V súčasnosti výučba týchto technológií prebieha zhruba v štyroch predmetoch/tematických okruhoch:

- *Internetové technológie*, kde sa študent oboznámi s technológiami HTML, XML, JavaScript.
- *Databázové systémy*, ktorý uvedie študenta do sveta relačných databáz a modelovania dát pomocou relácií.
- *Programovanie v OO jazyku*, výučbou moderného objektovo orientovaného programovacieho jazyka ako je C# alebo Java.
- *Programovanie webovských aplikácií*, kde sa predchádzajúce tri predmety prakticky využijú na výučbu tvorby JSP alebo ASP aplikácií.

Ako alternatívu k posledným krokom sa niekedy využíva tzv. LAMP stack, kde sa tieto predmety nahradia praktickou výučbou jazyka PHP.

Hlavným problémom súčasných prístupov je komplexita týchto technológií. Štandardným prístupom k odstráneniu komplexity vo vývoji softvéru je premiestniť bežné, no časovo náročné postupy do *softvérových knižníc* alebo komplexnejších vývojových balíkov tzv. *frameworkov*. Frameworkov na tvorbu webovských aplikácií existujú stovky, ako potvrdí krátky pohľad do Wikipédie [3]. Vybratie ideálneho frameworku pre začínajúcich študentov nie je

ľahké. Na jednej strane je trh dosť fragmentovaný, a nie je možné naučiť študentov všetky najpoužívanejšie frameworky. Na druhej strane, nie je možné vyučovať technológiu, čo ako krásnu, ktorú by študenti nemohli využiť ďalej v praxi.

My sme pri výbere frameworku zvažovali nasledujúce kritériá:

- pozvoľne stúpajúca krivka náročnosti,
- nabádanie k všeobecne odporúčaným vývojarským postupom,
- výber programovacieho jazyka,
- podpora viacjazyčnosti,
- integrácia,
- podpora viacerých platforiem,
- interoperabilita,
- expresívnosť a potenciál na rozširovanie,
- cena,
- použitie v praxi,
- materiály na výučbu.

Nie je možné v rozsahu tohto článku, a ani v schopnostiach autorov, zvážiť všetky možné frameworky a technológie použiteľné na webovské programovanie. Naším cieľom bude poukázať, že open-source framework web2py (<http://www.web2py.com>) je podľa týchto kritérií veľmi vhodným kandidátom na výučbu. Pri hodnotení sme vychádzali z našich doterajších skúseností s výučbou tradičných technológií (PHP, SQL, Java, .NET).

2 Framework web2py

Tento framework bol vyvinutý špeciálne na výučbu tvorby webovských aplikácií. Jeho hlavným autorom je Massimo Di Pierro, profesor informatiky na DePaul University v meste Chicago. Zdrojový kód frameworku je licencovaný pod GPL licenciou verzie 2, a k jeho vývoju prispel celosvetový tím vývojárov vrátane jedného z autorov [10]. Je vybudovaný na jazyku Python, a od väčšiny frameworkov sa líši poskytnutím DAL (database abstraction layer) komponentu, ktorý je používaný na udržiavanie modelu dát. Podobne ako LINQ technológia od Microsoftu, tento komponent umožňuje programátorovi uvoľniť sa z diktátu relačných databáz a pracovať s dátami bez toho, aby priamo používal SQL.

2.1 Výber jazyka

Existujú webovské frameworky pre jazyky od Erlang až po Perl. *Ak si môžete vybrať hociktorý jazyk, ktorý by ste si vybrali?* Táto rečnícka otázka má veľa odpovedí. Tu citujeme odpoveď jedného z prvých internetových milionárov. Paul Graham takto [9] vysvetľuje, prečo si vybrali jazyk Lisp pre ich firmu:

Vybrali sme Lisp. Jedným z dôvodov bol, že rýchly vývoj bol zjavne veľmi dôležitý na tomto trhu. Všetci sme začínali od nuly, a tak firma, ktorá vyvinie funkcionality skôr než konkurenčné, bude mať veľkú výhodu. Vedeli sme, že Lisp je veľmi dobrý jazyk na rýchly vývoj softvéru, a serverovské aplikácie ešte umocňujú tento efekt, pretože môžete publikovať softvér v tú istú minútu ako ho dokončíte.

Hoci výhody jazyka Lisp sú neodškriepiteľné, treba poznamenať, že je nerealistické očakávať od študentov bakalárskeho stupňa, aby sa naučili efektívne používať a oceniť také zložité koncepty jazyka Lisp ako je *continuation passing style programming*.

Alternatívnym pohľadom na výber jazyka je .NET stratégia firmy Microsoft. Táto nepredurčuje jeden jazyk (ako napríklad Java), ale umožňuje písať zdrojový kód s použitím množstva programovacích jazykov, ktoré sa všetky kompilujú do veľmi rýchleho .NET runtime. Samozrejme nevýhodou je, že aplikácie vyvinuté na .NET platforme bývajú s ťažkosťami prenositeľné z platformy Windows, a to napriek úsiliu firmy Novell vytvárať implementáciu .NET pre GNU/Linux a MacOS X.

Web2py je založený na jazyku Python. Hoci nie je najpoužívanejším jazykom, Python sa využíva v širokej škále aplikácií. Podľa Tiobe indexu [11] je to šiesty najpopulárnejší jazyk v júni 2009, predčiaci dokonca jazyk C# od spoločnosti Microsoft. Je to vhodný jazyk pre výučbu programovania, pretože podporuje rôzne programátorské paradigmy – imperatívnu, objektovo orientovanú ako aj funkcionálnu. Bez väčších problémov sa ho naučia aj začínajúci študenti. Je ľahko portovateľný a vďaka veľkému množstvu štandardných knižníc umožňuje tvorbu komplexných, interoperujúcich aplikácií na všetkých platformách. Používa slabšie typovanie než Java alebo C++ (táto téma a porovnanie s Javou sú bližšie rozobraté v [12]).

2.2 Internacionalizácia

V európskych podmienkach je podpora mnohojazyčnosti nevyhnutnou funkcionalitou vo webovom frameworku. Napriek tomu, podľa našich skúseností, správne fungovanie jazykov iných ako anglických je pre začiatočníkov tým najfrustrujúcejším aspektom vývoja aplikácií. Stratégia je zdanlivo veľmi jednoduchá – vstupné dáta, väčšinou v znakovej sade UTF-8, by sa mali bez straty mäččev a dĺžnov dostať do a späť z databázy. Napriek tomu, snáď kvôli veľkému množstvu kombinácií softwarových komponentov, cez ktoré sa reťazce presúvajú, je toto veľmi častým problémom. Web2py úspešne rieši tento problém, vrátane správneho prekódovania reťazcov do UTF-16 pre SQL Server.

Druhá časť internacionalizácie je podpora viacjazyčných rozhraní. V tejto oblasti web2py integruje komponent na tvorbu prekladov rozhraní do viacerých jazykov. Nevyhnutná zmena v kóde samotnej aplikácie je minimálna, stačí pridať volanie na `T()` konštruktor. Napríklad slovenskú aplikáciu reprezentovanú kontrolerom

```
def pozdrav():  
    return "Pe kny _ den"
```

možno zmeniť na viacjazyčnú jednoduchou zmenou

```
def pozdrav():
    return T("Pekny den")
```

a pridaním prekladov pre jednotlivé jazyky.

2.3 MVC

Tým, že web2py bol vytvorený viac než 10 rokov po nástupe svetového Internetu, jeho autor mal možnosť poučiť sa z problémov skorších frameworkov. A túto možnosť aj využil. Používa MVC (model-view-controller) postup popularizovaný frameworkom Ruby on Rails. Z výučbového hľadiska tento postup umožňuje študentom rozdeliť webovskú aplikáciu do troch samostatných častí:

- *model*, ktorý reprezentuje štruktúru dát,
- *kontroler*, v ktorom aplikačná logika manipuluje dáta,
- *pohľad* (angl. *view*), ktorý formátuje dáta do konečnej XHTML prezentácie.

Tu je príklad rozdelenia aplikácie na nahrávanie obrázkov na model, kontroler a pohľad.

```
# model
db=SQLDB('sqlite://storage.db')
db.define_table('image',
                db.Field('name'),
                db.Field('file', 'upload'))

# kontroler
def index():
    form = SQLFORM(db.image)
    if form.accepts(request.vars, session):
        response.flash = 'obrazok bol nahraty'
    return dict(form = form)

# pohlad
{{extend 'layout.html'}}
<h1>Upload obrazka </h1>
{{= form}}
```

Prevzatie kontroly dát a štruktúry dát webovskou aplikáciou z veľkej časti odstraňuje problém nazývaný 'object-relational impedance mismatch' [7]. Samozrejme, toto riešenie má svoju cenu, a tou je horšia integrácia s existujúcimi relačnými databázami. Napríklad, častou sťažnosťou na diskusných fórach je to, že web2py vyžaduje, aby tabuľky mali primárny kľúč nazvaný *id*. Na jednej strane takýto umelý kľúč môže byť považovaný za dobrú programátorskú praktiku. Na druhej strane je nerealistické očakávať od administrátorov existujúcich relačných databáz, aby takúto požiadavku splnili.

2.4 Riešenie problémov s SQL

Nedostatky jazyka SQL sú všeobecne známe, praktické aj teoretické (napr. [13]). Snáď najväčším praktickým je balkanizácia implementácií tohto jazyka, ktorá pod rúškom uvádzania nových funkcionalít prakticky znemožňuje zákazníkom prenos aplikácie z jedného systému riadenia bázy dát na druhý. Odtiaľ pochádza popularita zjednodušujúcich unifikujúcich technológií ako sú ODBC a Perl DBI, a odmietnutie komplexných technológií ako sú OLE DB.

PHP má snáď najväčší problém s SQL, pretože priam nabáda programátorov na vytváranie priestoru pre útoky pomocou SQL injection. Java je na tom lepšie, často využíva ORM frameworky ako je Hibernate na odbremenenie programátora od SQL jazyka. Takisto Microsoft s technológiou LINQ atakuje problém s SQL v aplikáciách, poskytujúc programátorom unifikovaný jazyk na manipuláciu dát z relačnej databázy, XML alebo pamätových štruktúr. Trend zakrývať variabilitu SQL je zjavný, a implementáciu DAL komponentu vo web2py možno považovať za prirodzené riešenie. Tento komponent umožňuje nielen zredukovať výučbu nevyhnutných relačných technológií z jedného semestra na jednu hodinu, ale takisto umožňuje použiť web2py na Google App Engine [1], ktorý nepoužíva relačnú technológiu.

2.5 Krivka náročnosti

Ako je napísané na wiki stránke web2py [5], tento framework sa líši od ostatných tým, že bol navrhnutý ako výučbový nástroj na DePaul University a preto prirodzene jeho krivka náročnosti je veľmi mierna. Je triviálne začať s ním pracovať (dokonca nepotrebuje ani inštaláciu, alebo administrátorské práva), a poskytuje webovské rozhrania na vývoj, debugovanie, testovanie, správu a administráciu bez použitia ďalších nástrojov. Pritom jeho celková veľkosť je približne 10 MB.

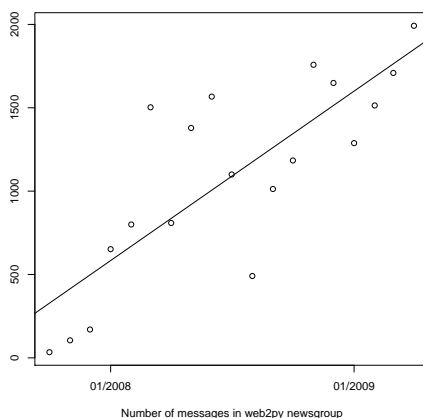
Jedným z hlavných problémov súčasných prístupov k výučbe je množstvo vedomostí, ktoré študent musí absorbovať, než je schopný ich skombinovať na tvorbu interaktívnej webovskej aplikácie. A pritom nie je principiálny dôvod, aby to tak bolo. Toto jasne demonštruje web2py na stránke [4] s 50 jednoduchými príkladmi, kde s minimálnym množstvom kódu (pozri Tabuľku 1) sa implementujú bežné úlohy. Táto jednoduchosť je podľa nášho názoru kriticky dôležitá pre výučbu.

2.6 Použitie v praxi

Na rozdiel od komerčného softvéru nie je možné vyhodnotiť použitie web2py v praxi v peňažnom meradle. Existujú však nepriame meradlá, ako je napríklad aktivita na diskusnom fóre web2py, ktorá je solídne rastúca. Obrázok 1 ukazuje nárast počtu príspevkov na fóre doplnený o priamku vypočítanú metódou lineárnej regresie. Poznamenávame, že nárast koncom roku 2008 je pravdepodobne následok slashdot efektu, keď web2py bolo priaznivo hodnotené na slashdot.com. Hoci tento efekt pominul, nárast príspevkov neprestáva.

Tab. 1: Počet riadkov v modeli (M), kontroleri (C) a pohľade (V) pre vzorové príklady zo stránky <http://www.web2py.com/examples/default/examples>

Príklad	M	C	V	Príklad	M	C	V
Hello world		2		Hello world (viacjazyčný)	2		
Hello world + pohľad		2	2	Zmena pohľadu	2		
HTML elementy		2		Hello world + oznam	3		
Zobrazenie request, session		2		Redirekcia	2		
HTTP výnimka		2		Python výnimka	3		
Zmena typu response		4		Použitie JSON	2		
Konvertovanie na RTF		9		Spracovanie RSS prúdu	20		
Konvertovanie WIKI syntaxe		9		Počítadlo pomocou session	4	6	
Zobrazenie premenných		1	4	Cyklus v šablóne	1	6	
if-then v šablóne		1	12	Výnimka v šablóne	1	8	
Funkcia v šablóne		1	7	Automatický escape	1	5	
Potlačenie escape		2	5	Automatické formátovanie	1	5	
Rozšírenie šablóny		6	3	Rozšírenie šablóny	6	3	
Základná šablóna		6	3	Formulár	14		
Model	34			Vloženie užívateľa	14	6	
Vloženie psa		7	6	Registrácia produktu	6	6	
Nákupný formulár		30	7	Odstránenie kúpy	3		
Zmena dát		3		Download obrázku	2		
Cache 1		4		Cache 2	4		
Cache 3		5		Cache 4	5		
Cache 5		5		Cache 6	6		
Cache 7		6		AJAX 1	8	12	
AJAX 2		3		AJAX 3	2	8	
Testovanie		8		Príklad prúdu	2		
XML-RPC		15					



Obr. 1: Počet príspevkov diskusného fóra web2py

2.7 AJAX

Pod termínom *Web 2.0* sa chápe nová generácia interaktívnych webovských aplikácií, ktoré sami iniciujú komunikáciu s webovským serverom. Typickým príkladom takejto aplikácie je Gmail od spoločnosti Google. Vývoj takýchto aplikácií je veľmi náročný, pretože vyžaduje kombináciu serverovských technológií spolu s programovaním v Javascripte na klientovi. Jedným, dobre známym riešením je využitie Google Web Toolkitu (GWT), ktoré je založené na programovaní aplikácie v jazyku Java. Menej známe je to, že existuje Pythonovská implementácia GWT nazvaná Pyjamas, a táto sa dá použiť na implementáciu AJAX vo web2py [6].

2.8 Integrácia

Programátori preferujú, ak všetky vertikálne komponenty platformy pochádzajú od jedného dodávateľa, či už je to Microsoft (Visual Studio + IIS + .NET framework) alebo autor Linuxovskej distribúcie (Eclipse + MySQL + PHP). Integračné testovanie vykonané týmto dodávateľom totiž ušetrí najmä laickejším užívateľom od ťažko riešiteľných problémov s interoperabilitou komponentov. V tejto oblasti web2py poskytuje bohatý balík, ktorý obsahuje webovský server cherrypy, systém riadenia bázy dát sqlite, konfiguračné riadenie, administratívne rozhranie vrátane lokalizačného rozhrania, ako i logovanie a zobrazovanie výnimok.

2.9 Dokumentácia

Na domovskej stránke www.web2py.com sa nachádza množstvo príkladov, tutoriálov a návodov ako pracovať s web2py. Aktívne je takisto diskusné fórum groups.google.com/group/web2py. Autor frameworku, Massimo DiPierro, je zároveň autorom knihy [8] o web2py, ktorú možno považovať za autoritatívnu referenciu. V papierovom vydaní sa dá kúpiť za 50 €, v elektronickej forme za 10 €. Tu podávame stručný prehľad obsahu.

Knihy má 256 strán a je členená do 10 kapitol. Po všeobecnej úvodnej kapitole nasleduje 20 stranová kapitola o jazyku Python. Najdôležitejšie na prácu s web2py sú kapitoly 3 až 7.

V tretej kapitole (40 strán) autor krok za krokom buduje aplikácie, aby prakticky ukázal používanie frameworku. Aplikácie sú rastúcej komplexity, od jednoduchej aplikácie na výpis reťazca, cez blogovaciu aplikáciu s obrázkami až po WIKI aplikáciu. Štvrtá kapitola (40 strán) vysvetľuje celkovú štruktúru web2py, ako sa mapujú URL, a základné objekty: request, response, session, cache, internacionalizáciu a všeobecný workflow. V piatej kapitole sú vysvetlené šablóny na integráciu HTML s Pythonom. Šiesta kapitola sa zaoberá DAL, komponentom na prácu s databázami a siedma kapitola preberá webovské formuláre.

V ôsmej kapitole sa autor venuje implementácii AJAX, v deviatej ukazuje ako uvádzať web2py aplikácie do prevádzky vrátane Google App Engine, a desiatu adresuje rôzne webovské technológie ako sú XML-RPC, JSON, RSS/ATOM, RTF, Flash, Email apod.

3 Porovnanie s PHP

Je všeobecne známe, že programovanie s PHP, napriek jeho rozšírenosti, je náročné (napr. [2]). V tejto časti článku ukážeme priame porovnanie web2py so Zend frameworkom, ktorý je tiež navrhnutý ako MVC framework. Na porovnanie sme implementovali kompletnú blogovaciu aplikáciu vo web2py aj Zende.

3.1 Porovnanie modelu

V Zend frameworku sa používa externá databáza a preto programátor potrebuje na rozdiel od web2py znalosti o vytváraní relačných databáz. Kód vo web2py bol približne o 50% kratší.

3.2 Porovnanie kontroleru

Každý kontroler v Zende musí byť definovaný ako trieda odvodená od `Zend_Controller_Action`. Na vytvorenie formulára na editáciu sa používa metóda `addElement()`. Spracovanie tohto formuláru vyžaduje manuálne kódovanie, zatiaľ čo vo web2py sa formulár vytvorí aj spracuje v niekoľkých riadkoch kódu.

3.3 Porovnanie pohľadu

Vo frameworku Zend sa vytvára HTML stránka skombinovaním statického obsahu s výsledkom PHP fragmentov. V porovnaní s týmto prístupom, web2py vedie študenta k vytváraniu správne formovaných XHTML stránok tým, že preddefinuje triedy pre HTML elementy. Toto má za nepriamy následok prevenciu cross-site útokov na webové aplikácie, čo je rizikom pre menej skúsených programátorov.

4 Záver

V tomto článku sme sa snažili prezentovať argumenty, ktoré nás viedli k rozhodnutiu používať web2py na výučbu študentov na našej univerzite. Pre nás najdôležitejším faktorom bol fakt, že sa jednoducho implementujú jednoduché webové stránky, no popritom študent nie je obmedzený v možnosti rozširovať webovskú aplikáciu. Hoci sa web2py odlišuje v niektorých aspektoch od momentálne najpoužívanejších frameworkov, na ktoré študent narazí v praxi, veríme, že je veľmi dobre použiteľný pri výučbe webového programovania, a to najmä v nasledujúcich prípadoch

- pri výučbe na stredných školách, kde je obmedzený hodinový rozsah
- na vysokých školách ako publikačnú technológiu pre neinformatikov
- ako open source alternatíva k tradičným Java/.NET frameworkom
- ako aplikáciu v rámci výučby jazyka Python

Za hlavný nedostatok z hľadiska výučby možno zaradiť nedostatok študijných materiálov o web2py v slovenskom jazyku.

Literatúra

- [1] Google App Engine <http://code.google.com/intl/sk/appengine/>
- [2] Blurring of MVC lines: Programming the Web Browser <http://advogato.org/article/993.html>
- [3] Comparison of web application frameworks, http://en.wikipedia.org/wiki/List_of_web_application_frameworks
- [4] web2py Examples, <http://www.web2py.com/examples/default/examples>
- [5] web2py, <http://en.wikipedia.org/wiki/Web2py>

- [6] Using pyjamas with web2py, <http://mdp.cti.depaul.edu/AlterEgo/default/show/203>
- [7] Object-relational impedance mismatch, http://en.wikipedia.org/wiki/Object-relational_impedance_mismatch
- [8] DI PIERRO, M.: *Web2Py Manual*, Wiley 2008; <http://www.lulu.com/content/4968879>
- [9] Want to start a startup? <http://www.paulgraham.com/avg.html>
- [10] web2py Developers, <http://www.web2py.com/examples/default/who>
- [11] TIOBE Programming Community Index, <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [12] Strong versus Weak Typing, A Conversation with Guido van Rossum, <http://www.artima.com/intv/strongweakP.html>
- [13] HUGH, D. – DATE, C. J. *The third manifesto*. ACM SIGMOD Record (New York, NY, USA: ACM Press) 24 (1): 39–49, (March 1995).

Kontaktná adresa

Ján KARABÁŠ, (Mgr., PhD.),
Univerzita Mateja Bela, Ďumbierska 1,
974 11 Banská Bystrica,
karabas@savbb.sk

Jozef SILÁČI, (Mgr.),
Katedra informatiky, Univerzita Mateja Bela, Tajovského 40,
974 11 Banská Bystrica,
silaci@fpv.umb.sk

Ondrej ŠUCH, (PhD.),
Katedra informatiky, Univerzita Mateja Bela, Tajovského 40,
974 11 Banská Bystrica,
such@fpv.umb.sk

**Fakulta riadenia a informatiky
Žilinská univerzita**

**OTVORENÝ SOFTVÉR VO VZDELÁVANÍ,
VÝSKUME A V IT RIEŠENIACH**



**Zborník príspevkov medzinárodnej konferencie
OSSConf 2009**

**2.–5. júla 2009
Žilina, Slovensko**