

▣ Dobre prakse uporabe in razvoja razširitev v sistemu Joomla!

Jaka Kužnik, Gregor Polančič

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Inštitut za informatiko, Smetanova 17, 2000 Maribor
jaka.kuznik@gmail.com; gregor.polancic@uni-mb.si

Izvleček

V zadnjih letih smo priča porastu uporabe sistemov za upravljanje spletnih vsebin, izmed katerih spada odprtokodni sistem Joomla! med najzmogljivejše in najpopularnejše. V prispevku je predstavljen omenjeni sistem, skupaj s triki in nasveti, ki nam olajšajo delo z njim. Pri tem smo izpostavili vidik varnosti in podali načine, kako se izogniti varnostnim tveganjem. Ker je Joomla! zelo prilagodljiv sistem, smo poseben razdelek posvetili značilnostim in izzivom, ki se pojavijo pri uporabi in implementaciji njenih razširitev. V prispevku so podani primeri izdelave lastnih gradnikov, s katerimi si lahko na preprost način prilagodimo izgled in funkcionalnosti spletnega mesta. Ugotavljamo, da smo pri sistemu Joomla! omejeni predvsem z lastno ustvarjalnostjo.

Ključne besede: sistem za upravljanje spletnih vsebin Joomla!, dinamična spletna mesta, razširitve, predloga, komponenta, modul, vtičnik, splošna javna licenca.

Abstract

Best Practices of Usage and Extensions Development in Joomla!

In the last years we have been witnesses to the rise of web content management systems, among which Joomla! is considered as one of the most powerful and popular. In this article, we have presented Joomla!, its characteristics, license policy and common problems. Based on common problems, we defined best practices and how to avoid such problems. Since security is a crucial aspect of websites, we focused on Joomla!'s security problems and their workarounds. In the following chapter, we explained the importance of extending Joomla! to fulfill user requirements. We presented different types of Joomla! extensions and proposed best practices for their successful and simple implementation. We found that Joomla! offers plenty of extensibility possibilities. If properly used, we are limited mainly by our own creativity.

Keywords: Joomla! Content Management System, dynamic web sites, extension, template, components, modules, plug-in, GNU General Public License.

1 UVOD

Zaradi vse večjega povpraševanja po enostavnih programskih rešitvah za razvoj spletnih mest (angl. website), smo v zadnjih letih priča vedno večji ponudbi komercialnih (plačljivih) in odprtokodnih (brezplačnih) sistemov za upravljanje s (spletnimi) vsebinami (angl. Content Management Systems, v nadaljevanju CMS). CMS-i so orodja za izdelavo in urejanje dinamičnih spletnih mest, ki težijo k temu, da lahko brez programerskih znanj implementiramo in upravljamo z vsebino, kot tudi s celotnim spletnim mestom. Zaradi tega omogočajo zmanjšanje stroškov izdelave, urejanja in vzdrževanja spletnih mest, kot tudi rešitev za hitro in enostavno objavo ažurnih informacij. Posamezniki ali podjetja se lahko z odprtokodnimi CMS-i delno ali v celoti izognejo stroškom vzdrževalnih pogodb in stroškov programiranja (Sreves 2009; Travis 2011; Slovenska skupnost uporabnikov Joomla CMS).

CMS-i omogočajo preprosto in celovito upravljanje spletnih mest tehnično nepodkovanim uporabnikom, saj urejanje vsebin ni pogojeno s poznavanjem programskih jezikov, označevalnih jezikov in drugih spletnih tehnologij. Z uporabo brskalnika lahko uporabnik CMS-a preprosto spreminja vsebino svoje spletne strani kadar koli in od kjer koli. Preprosto lahko dodajamo nove menije in podstrani, katere niso bile predvidene pri prvotni izdelavi in so se pojavile kasneje. Določimo lahko, kdaj naj se določena vsebina objavi in kako dolgo naj se prikazuje na spletni strani, vodimo statistike pregleda in upravljamo s skritimi vsebinami. CMS-ji imajo možnost oblikovanja metaoznak, ki so pomembne za umestitev spletnih strani v iskalnike. Značilnost

CMS-ov je, da na eni strani omogočajo centralizirano in konsistentno vzdrževanje spletnih mest, medtem ko po drugi strani omogočijo decentralizacijo delovnih opravil (npr. v vzdrževanje in objavljanje vsebin spletnega mesta lahko vključimo svoje prijatelje, sodelavce oz. kogar koli, ki je kakor koli povezan s spletnimi stranmi). S tem nam CMS omogoča, da imajo spletni obiskovalci vedno na razpolago sveže in aktualne informacije (Rahmel 2007; Rahmel 2009; Kramer 2011).

Med pglavitne slabosti CMS-ov spada težavno prilagajanje specifičnim zahtevam uporabnikov ali celo nezmožnost le-tega (npr. specifična funkcionalnost, ki je CMS ne podpre ali specifičen dizajn, ki ga

ni mogoče do potankosti realizirati v CMS-u). Omejeno težavo naslavlja različni CMS-i različno in so pri tem različno uspešni. V prispevku bomo predstavili in raziskali enega najbolj razširjenih sistemov – sistem Joomla!. Pri tem se bomo osredinili prav na dobre prakse uporabe in na razvoj lastnih gradnikov v sistem Joomla!, ki omogočajo, da ta CMS prilagodimo še tako eksotičnim uporabniškim zahtevam.

Iz tabele 1 je razvidno, da ima vsak CMS svoje prednosti in slabosti. Tako je izbira odvisna od naših potreb in zahtev. Pred končno izbiro sistema CMS je treba preizkusiti delovanje in uporabnost več različnih CMS-ov in preveriti izkušnje drugih uporabnikov z njimi (Quinn & Gardner-Madras 2010).

Tabela 1: **Primerjalna tabela priljubljenih odprtokodnih CMS-ov (Quinn & Gardner-Madras 2010)**

	WordPress	Joomla! 1.6	Drupal 7.0	Plone
Enostavnost gostovanja in namestitve	3	3	3	1
Enostavnost izdelave: enostavne strani	3	2	2	1
Enostavnost izdelave: kompleksne strani	3	3	2	2
Enostavnost uporabe: uredniki	3	2	2	3
Enostavnost uporabe: administrator	3	2	2	2
Grafična prilagodljivost	3	3	3	3
Dostopnost in optimizacija spletne strani	2	3	1	3
Strukturna prilagodljivost	2	2	3	3
Vloge uporabnikov in potek dela	1	3	2	3
Skupnost/splet 2.0 funkcionalnosti	3	2	3	2
Razširitev in združevanje	3	3	3	3
Varnost	1	2	2	3
Podpora/moč skupnosti	3	3	3	3
Skupaj	33	33	31	32

Odlično = 3 Dobro = 2 Zadostno = 1 Ni podprto = 0

Za preizkušanje in testiranje odprtokodnih sistemov je na voljo spletno mesto *The Open Source CMS* na naslovu <http://www.opensourcecms.com>. Na tej strani lahko brez nameščanja programske opreme preverimo delovanje najrazličnejših sistemov CMS, ki so temeljijo na PHP/MySQL. Prav tako lahko na spletni strani *The CMS Matrix* <http://www.cmsmatrix.org> medsebojno primerjamo več kot tisoč sistemov CMS.

Sistem Joomla! smo izbrali zaradi njegove zmožljivosti, razširjenosti, široke mednarodne podpore, dobrih referenc in podpore v slovenskem jeziku. Zaradi svoje mednarodne razpoznavnosti ima Joo-

mla! zagotovljen nadaljnji razvoj, s katerim upravičuje uporabo v poslovne namene. Joomla! s svojo fleksibilnostjo omogoča pokrivanje različnih tipov spletnih mest in zahtev uporabnikov oziroma naročnikov. Uspešno se uporablja za različna osebna in poslovna spletna mesta, spletne trgovine, portale za šole, univerze, državno upravo, korporativne portale, podporo medijskim hišam, raznim neprofitnim organizacijam in intranetnim stranem (Juvančič & Juvančič 2006; Skrt 2006).

Na spletnih straneh Open Source CMS Award (Packt publishing) v tej kategoriji programske opre-

me, podeljujejo nagrade najboljšim CMS-om. Sistem Joomla! je za leto 2011 ponovno prejel priznanje za najboljši odprtokodni CMS. Tabela 2 prikazuje rezultate zadnjih šestih let.

V nadaljevanju je prispevek organiziran takole: v drugem razdelku je podrobneje predstavljen sistem Joomla!. Izpostavljena je njegova zmogljivost, ki ne omejuje uporabnikov, temveč jim omogoča neomejene možnosti nadgradnje in prilagajanja spletnih mest z razvojem lastnih gradnikov. V tretjem razdelku so predstavljene rešitve za pogoste probleme, na katere naletijo skrbniki oz. upravljavci sistema Joomla! pri vzpostavitvi spletnega mesta. Analizirani so vzroki pogostih težav, kje se te pojavijo najpogosteje, in predlagane rešitve. Poseben poudarek je na zagotavljanju varnosti spletnega mesta, ki temelji na sistemu Joomla!. V četrtem razdelku so predstavljene vrste in načini izdelave lastnih gradnikov, s pomočjo katerih se lahko Joomla! prilagodi specifičnim uporabniškim zahtevam. Sklepne misli so podane v zadnjem razdelku.

Tabela 2: **Zmagovalci The Open Source CMS Award (Packt publishing)**

Leto	1. mesto	2. mesto	3. mesto
2006	Joomla!	Drupal	Plone
2007	Drupal	Joomla!	CMS Made Simple
2008	Drupal	Joomla!	DotNetNuke
2009	WordPress	MODx	SilverStripe
2010	CMS Made Simple	SilverStripe	MODx
2011	Joomla!	Drupal	Plone

2 PREDSTAVITEV SISTEMA JOOMLA!

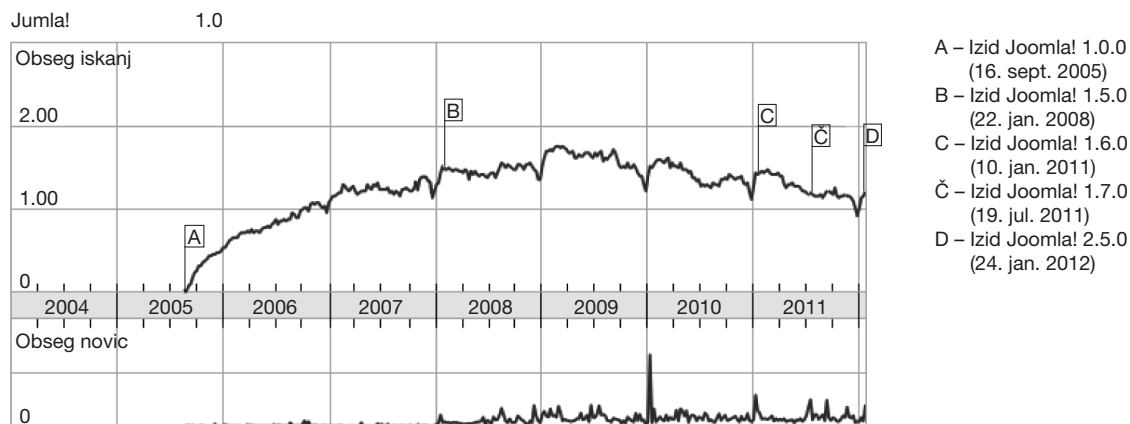
Sistem Joomla! je nastal leta 2005 iz sistema CMS Mambo, ko se je del programerjev odcepil od projekta Mambo in nadaljeval razvoj pod drugim imenom. Ime Joomla! izhaja iz afriškega jezika svahili in se izgovarja džumla. Beseda pomeni »skupnost«, »kot celota« in »vsi skupaj«. Iz tega pomena izhaja tudi logotip (slika 1), ki predstavlja skupnost, povezano v celoto. Sestavni del imena je tudi klicaj (Rahmel 2007).



Slika 1: **Logotip sistema Joomla!**

Za organizacijsko, finančno in pravno podporo odprtokodnega projekta Joomla! stoji neprofitna organizacija Open Source Matters, ki se nahaja v ZDA. Joomla! je nastala kot alternativa plačljivim CMS-om, ki pogosto pomenijo prevelik finančni zalogaj za neprofitne organizacije in manjša podjetja, ki si želijo sama ustvariti in vzdrževati spleta mesta. Joomla! je zelo močno in robustno orodje, ki ga uporablja vedno več ljudi. Njegova univerzalnost ga je povzdignila med najbolj priljubljene CMS-e (North 2009; Marriott & Waring 2010).

Razširjenost sistema Joomla! lahko preverimo v orodju Google trends. Na spodnjem grafu (slika 2) vidimo obseg iskanj v iskalniku Google, ki prikazuje kvocient dejanskega števila iskanj s povprečnim številom iskanj v trenutnem obdobju. Črke (A, B, C, Č, D) v grafu prikazujejo izid različnih verzij sistema Joomla!.



Slika 2: **Obseg iskanj besede Joomla! (Google trends)**

Joomla! temelji na priljubljeni arhitekturi LAMP, ki jo sestavljajo operacijski sistem Linux, spletni strežnik Apache, podatkovna baza MySQL in programski jezik PHP. S tem programskim jezikom lahko sistem Joomla! tudi prilagajamo in razširjamo. Na spletu najdemo številne brezplačne, odprtokodne in komercialne razširitve, ki povečajo nabor funkcionalnosti in izgled sistema Joomla!. Izhodiščna stran za razvijalce in uporabnike sistema Joomla! je <http://www.joomla.org>. Najbolj uporabne poveza-ve si lahko ogledamo v tabeli 3 (Skrť 2006; Shreves 2009; Kramer 2010).

Tabela 3: **Uradna spletna mesta Joomla!**

Ime	URL
Joomla! (glavna stran)	http://www.joomla.org/
Joomla! Code	http://joomlancode.org/
Joomla! Developer Site	http://developer.joomla.org/
Joomla! Extensions Directory	http://extensions.joomla.org/
Joomla! Documentation	http://docs.joomla.org/
Joomla! Forums	http://forum.joomla.org/

Tabela 4: **Pozitivni in negativni vidik sistema Joomla! (Shreves 2009)**

Pozitivni vidik	
Več kot 10 mio prenosov	Številka pomeni rastoč projekt z veliko uporabniki.
Na voljo več kot 4,500 razširitev	Veliko število razširitev pomeni, da lahko vaši strani dodajate različne funkcionalnosti.
Uporablja sistem LAMP	Preprosto najdemo gostovanje in pomoč.
Široka podpora	Na spletu najdemo veliko podpore.
Popolna dokumentacija	Na uradni strani sistema Joomla! je dosegljiva popolna dokumentacija.
Aktivna skupnost	Aktivna in dinamična skupnost pomeni, da lahko dobimo podporo na forumih ali sodelujemo pri diskusijah.
Negativni vidik	
Vzdrževanje prilagojenih strani	Pri prilagajanju določenih delov strani lahko prihaja do težav, ko hočemo prilagojeni del nadgraditi na najnovejšo verzijo.
Ne deluje brez povezave.	Če spletno mesto potrebuje delovanje brez internetne povezave, Joomla! ni rešitev. Sistem ne vključuje stroja za podporo delovnim tokovom (angl. workflow engine).

3 ANALIZA POGOSTIH PROBLEMOV UPORABE SISTEMA JOOMLA!

V nadaljevanju so predstavljeni predlogi in rešitve za najpogostejše probleme, na katere naletijo skrbniki sistema Joomla!. Obravnavane so težave, ki se pogosto pojavljajo na različnih slovenskih in tujih forumih Joomla! (Slovenska skupnost uporabnikov Joomla CMS; Joomla Templates, Joomla Extensions for the Joomla CMS; Joomla! Forum). Poseben poudarek je na povečanju zaščite spletnega mesta.

Čeprav je sistem Joomla! uspešen v najrazličnejših primerih uporabe, lahko še vedno naletimo na primer, kjer se ne bo obnesel najbolje. Zato je treba pred odločitvijo »za« ali »proti« ovrednotiti prednosti in omejitve sistema Joomla!, ki so povzete v tabeli 4.

Sistem Joomla! je licenciran s splošno javno licenco GNU (angl. GNU General Public License, v nadaljevanju GNU/GPL). S to licenco lahko predelamo in prodamo delo pod istimi pogoji, ki se nanašajo na izvorno kodo oz. pravico, da jo kupec prejme in distribuira naprej. Pogoje moramo upoštevati za ves program, tudi v primeru, ko je bilo uporabljenih le nekaj vrstic kode, izdane pod licenco GNU/GPL. V primeru, da ne želimo programa distribuirati pod to licenco, čeprav del kode uporabimo v svojem programu, moramo GNU/GPL kodo dati v samostojen program, katerega potem iz svojega programa kličemo s posamezno funkcijo. Kodo GNU/GPL v svoj program lahko nato pokličemo iz samostojnega programa (The GNU Operating System; Center odprte kode Slovenija; Creative Commons; Creative Commons Slovenija).

3.1 Slovenski jezik

Podpora slovenskemu jeziku je pri izdelavi slovenskega spletnega mesta zelo pomembna, saj imamo v nasprotnem primeru veliko težav s posebnostmi slovenskega nabora znakov – s šumniki. Te težave se pojavljajo tako pri prikazu vsebine, kot tudi pri zapisih podatkov v podatkovno bazo. Sistem Joomla! je v svoji zgodovini imel nemalo težav s šumniki predvsem v prvih verzijah, saj je bila to največkrat zasto-pana težava na slovenskih forumih Joomla! (Slovenska skupnost uporabnikov Joomla CMS).

Jezikovne podpore za sistem Joomla! so izdane v obliki razširitev, ki jih preprosto namestimo na svoje spletno mesto. Osnovna verzija sistema Joomla! pozna tri različne jezikovne razširitve:

- slovensko jezikovno podporo pri namestitvi sistema Joomla!,
- slovensko jezikovno podporo za »ozadje«¹ sistema Joomla!,
- slovensko jezikovno podporo za »ospredje«² sistema Joomla!.

Obstajajo tudi zahtevnejše komponente, ki imajo svojo jezikovno datoteko. Uradne jezikovne razširitve najdemo na spletni povezavi <http://community.joomla.org/translations.html>. Običajno so za slovenski jezik na voljo tudi na strani slovenske skupnosti Joomla! – SloJoomla (Slovenska skupnost uporabnikov Joomla CMS).

3.2 Optimizacija sistema Joomla!

Za optimizacijo spletnih mest za namene boljše uvrstitve v spletnih iskalnikih (angl. Search Engine Optimization – SEO), ki temeljijo na sistemu Joomla!, se uporabljajo iste tehnike kot pri drugih statičnih straneh ali straneh, narejenih v dinamičnih okoljih. Joomla! se je v svojih različicah nadgrajeval tudi na tem področju, tako da lahko marsikaj naredimo iz skrbniškega dela strani. Za boljšo vidnost naše strani lahko upoštevamo tale priporočila (SEO training):

- dodamo metaopis in ključne besede strani,
- dodamo metaopis in ključne besede vsakemu članku na naši strani,
- pravilno izberemo naslove strani in vsebine,
- optimiziramo vsebine strani, da se zelene ključne besede na določeni strani pojavljajo v pravšnjem številu. Prav tako uporabimo čim več povezav na preostale objavljene članke,
- za izdelavo strani uporabljamo urejevalnike WYSIWYG³, ki delajo veljavno kodo XHTML.

Napadalci običajno iščejo ranljive točke na internetu preko iskalnika Google z ukazom »inurl:«, s katerim iščejo posamezne besede, oz. nize znakov v spletnih naslovih vira (angl. Uniform Resource Locator, v nadaljevanju URL). Joomla! ima že v namestitveni različici vgrajeno osnovno optimizacijo spletnih strani, ki med drugim poskrbi za prijaznejše spletne naslove (angl. Search Engine Friendly, v na-

daljevanju SEF), kar pomeni, da so naslovi lažje dostopni, da se lažje indeksirajo in da jih hitreje najdejo pajki iskalnikov. Tako se prepíše naslov URL in odstrani vse informacije, ki se prenašajo preko naslova. Naslovi SEF so lažje berljivi, izboljša se tudi uvrstitev spletnih strani v različnih iskalnikih (North 2009; Nasvet). Optimizacija SEF tako naredi iz zapisa URL, kot je npr.

http://ime-domene.si/index.php?option=com_content&view=article&id=1&Itemid=102,

prijaznejši zapis, tako za iskalnike, kot tudi za uporabnike:

<http://ime-domene.si/test.html>.

Za zahtevnejše uporabnike so na voljo razširitve SEF in SEO, ki jih najdemo na straneh z razširitvami Joomla!.

3.3 Varnost

Spletni uporabniki, ki poznajo rek »100-odstotna varnost ne obstaja«, se dobro zavedajo, da na spletu obstajajo različna varnostna tveganja. Ker je spletno mesto vedno na spletu, je zagotavljanje spletne varnosti trajen proces in ne samo trenutno stanje.

Varnost spletnega mesta Joomla! lahko povečamo s smernicami, ki so opisane v tem poglavju. Pomembno je, da smo na varnost spletnega mesta pozorni že ob namestitvi sistema Joomla!. Napadalci običajno iščejo ranljive točke, ki so povezane s specifičnim izvajalnim okoljem in specifičnimi verzijami sistema Joomla!, zato je priporočljivo uporabiti nestandardne predpone tabel, odstraniti nepotrebne razširitve, podatke in številke različic. Prav tako je treba omejiti dostope do map, datotek in skrbniškega dela. Priporočamo uporabo datoteke »htaccess«⁴ in iskalniku prijaznih URL-jev. Pomembna je tudi uporaba močnih gesel in redna izdelava varnostnih kopij (Marcofolio.net).

3.3.1 Neuporabljane razširitve in podatki

Na objavljenem spletnem mestu Joomla! je pomembno, da imamo le gradnike, ki jih potrebujemo za normalno delovanje. Tudi če podatki niso objavljeni ali je razširitev onemogočena, lahko njene datoteke še vedno pomenijo ranljivost sistema, ker so še vedno

¹ Ozadje (angl. backend) – skrbniški del spletnega mesta.

² Ospredje (angl. frontend) – uporabniški del spletnega mesta.

³ WYSIWYG (angl. What You See Is What You Get) – kar vidiš, boš tudi dobil.

⁴ Datoteka .htaccess je namenjena izboljšanju varnosti, prepoveduje nasilen vdor v CMS.

na strežniku. Že v sami namestitvi sistema Joomla! imamo možnost namestiti vzorčne podatke, ki so namenjeni učenju in spoznavanju le-tega. Vzorčni podatki za delovanje spletne strani niso potrebni in nam lahko škodujejo, saj po tej vsebini napadalec ugotovi, da je spletna stran narejena v sistemu Joomla!.

Prav tako je priporočljivo, da se odstranijo vse komponente, moduli, vtičniki, predloge in druge razširitve (predstavljene so v četrtem razdelku), ki jih ne uporabljamo. Onemogočimo vse funkcionalnosti, ki jih trenutno ne potrebujemo, npr. če ne potrebujemo registracije in prijave uporabnikov, onemogočimo registracijo uporabnikov. Tudi neuporabljene predloge in slike lahko predstavljajo varnostno grožnjo, npr. vsaka datoteka lahko izda nepridipravo informacije o spletnem mestu oz. o razširitvi in tako ta lažje najde ranljive točke spletnega mesta. Zato jih je priporočljivo odstraniti.

Kot smo že omenili, je pomembno, da tako stran kot tudi nameščene razširitve sledijo nadgradnjam in novejšim različicam, s katerimi so običajno odpravljene varnostne grožnje. Priporočljivo je, da odstranimo prikaz števil različic, saj se večina ranljivosti nahaja v natančno določenih verzijah sistema Joomla!. Tako zmanjšamo možnost, da napadalci identificirajo ranljivo verzijo sistema Joomla! in izdelajo načrt za napad, ki je značilen za specifično verzijo.

3.3.2 Dostop do map in datotek

Pomembno je, da vse datoteke in mape zaščitimo pred možnostjo pisanja. CHMOD⁵ oz. pravice map najpogosteje določimo preko odjemalca FTP, tako da označimo mapo, kliknemo lastnosti in določimo pravice. Priporočljivo je nastaviti pravice map na 755, datoteke na 644, datoteko *configuration.php*⁶ pa zaradi pomembnih podatkov na 444. Večje pravice dodelimo le, kadar skripta zapisuje v določeno datoteko ali direktorij.

Dnevniška datoteka (angl. log) in predpomnilnik imata privzeto lokacijo. Z vidika varnosti jo je priporočljivo prestaviti zunaj lokacije spletnega mesta, kjer je težje dostopna. Iz zapisov, ki se nahajajo v dnevniški datoteki in predpomnilniku, lahko napadalec pridobi koristne informacije za napad na spletno mesto.

Datoteka *.htaccess*, ki je namenjena izboljšanju varnosti, vsebuje določene kode, ki prepoveduje-

jo nasilen vdor v CMS. Privzeto je onemogočena, s preimenovanjem datoteke *htaccess.txt* v *.htaccess* jo omogočimo. Vključena mora biti tudi ob uporabi SEF-a. Privzeta datoteka *.htaccess* nam ponuja nekaj varnosti, seveda pa jo lahko še izboljšamo. Uporablja se za preusmeritve, zaklepanje dostopov, nastavljanje lepših naslovov URL idr. (Snipt).

3.3.3 Skrbniški račun

Sistem Joomla! ima poznan varnostni problem z dostopom do skrbniškega dela strani, saj lahko vsak uporabnik preprosto preveri, ali je spletno mesto ustvarjeno s sistemom Joomla!, tako da v brskalniku doda ime podmape »administrator« (npr. *www.naslov-strani.si/administrator*). Zato je priporočljivo zamaskirati skrbniški del. Ena izmed rešitev je vtičnik »Secure Authentication«, ki preprečuje dostop do skrbniškega dela strani z oblikovanjem naslova URL z ustreznim ključem.

Privzeti skrbniški račun, ki ga oblikujemo ob namestitvi spletnega mesta, ima privzeto uporabniško ime »admin« in privzeto identifikacijsko (id) številko računa 42. Za povečanje varnosti je priporočljivo skrbniški račun zamenjati z novim, privzetega pa izbrisati.

Drug varnostni problem lahko nastopi, če pozabimo geslo »super administratorja«. ⁷ Takšna izguba povzroči skrbnikom veliko nevšečnosti, pogosto se odločijo za kreiranje novega spletnega mesta. Vendar obstaja elegantnejša rešitev. Joomla! zapisuje gesla v kodiranem formatu MD5, ki iz niza znakov naredi 128-bitno število, obratna pot pa ni mogoča (razen s poskušanjem). Tako ne moremo videti, kakšno je geslo, lahko ga le spremenimo. Za ustrezno kodo MD5 je najbolje uporabiti temu namenjen program. To kodo nato s pomočjo orodja PhpMyAdmin zapišemo v skrbniškemu računu in geslo je spremenjeno.

4 DOBRE PRAKSE RAZVOJA RAZŠIRITEV JOOMLA!

Dobre prakse so nastale kot odgovor na probleme, ki smo jih predstavili v tretjem razdelku. So rezultat uporabe sistema Joomla! v več projektih razvoja dinamičnih spletnih mest, v okviru katerih smo pogosto in ponavljajoče probleme začeli reševati podobno. Dobre prakse so se torej izoblikovale na lastnih izkušnjah in priporočilih drugih razvijalcev, ki smo jih prevzeli iz spletnih mest Joomla! (tabela 3).

⁵ CHMOD – ukaz se uporablja za spreminjanje pravic dostopa do datotek ali map.

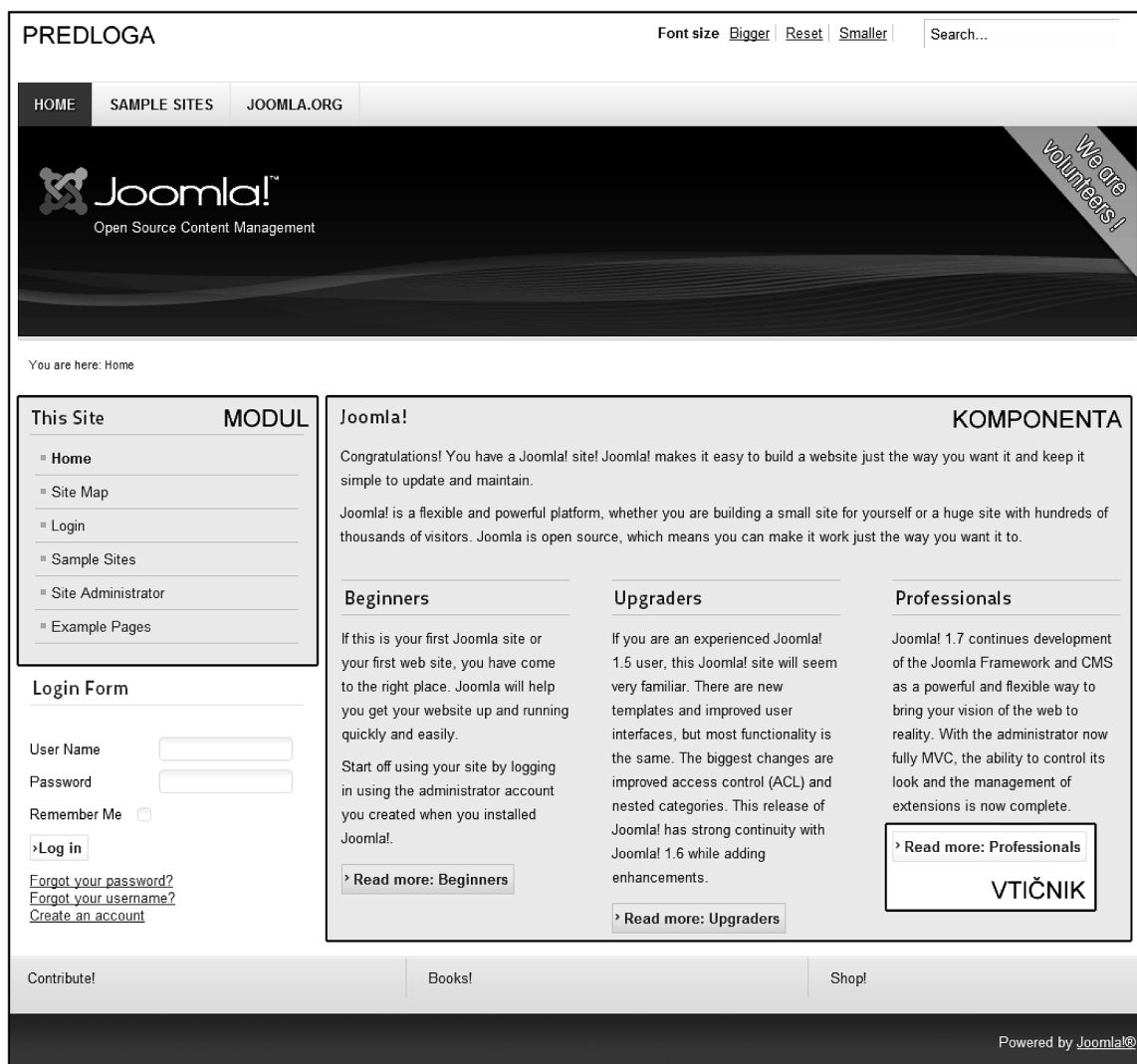
⁶ *configuration.php* – datoteka vsebuje globalne nastavitve spletnega mesta.

⁷ Super administrator – uporabniška skupina z najvišjimi pravicami.

Sistem Joomla! je zgrajen modularno in že v osnovi omogoča izdelavo zmogljivih in prilagodljivih spletnih mest. Poleg tega lahko sposobnejši in zahtevnejši uredniki na spletu najdejo številne razširitve Joomla!, ki prilagodijo ali razširijo izgled in funkcionalnost spletnega mesta. Večina razširitev je

objavljenih pod licenco GNU, nekaj jih je tudi plačljivih. Najpogostejše uporabljene razširitve so galerije slik, večpredstavnosti, dinamični obrazci, forumi, koledarji, spletne trgovine ipd.

Joomla! pozna te vrste razširitev: komponente, module, vtičnike, jezikovne pakete in predloge (slika 3).



Slika 3: Na sistemu Joomla! temelječa spletna stran z označenimi razširitvami

Komponente so najkompleksnejši in bistveni tip razširitev. So kot majhne aplikacije, ki običajno vplivajo na glavni, osrednji del strani. Moduli so manj kompleksni kot komponente in jih za razliko od komponent lahko postavimo na podstrani v poljubnem številu. Vtičniki so posebni deli kode, ki delujejo na vso stran skozi vse komponente in module. Vtičniki se uporabljajo na straneh in imajo nalogo oblikovanja

izhoda komponente ali modula pred prikazom. Predloge Joomla! zagotavljajo videz in stil spletnih strani ter upravljajo grafični dizajn, kar vključuje barve, grafiko in tipografijo.

Pred izdelavo lastne razširitve je dobro poznati prednosti oz. omejitve vsake vrste razširitev, saj se s tem izognemo nepravilni izbiri in implementaciji. Za večino ranljivosti sistema Joomla! so krive razširitve,

saj jih najdemo na spletu v precejšnjem številu. Mnoge so napisane površno in brez razmišljanja o varnosti. Zato moramo biti zelo pozorni, kakšno razširitev nameščamo na svojo spletno stran, dodatna previdnost pa je potrebna pri neuradnih razširitvah Joomla!. Najpreprosteje preverimo razširitev z obiskom razvijalčeve spletne strani, pregledamo mnenja uporabnikov forumov, preverimo, kako hitro odpravljajo napake itd.

V nadaljevanju so predstavljene dobre prakse razvoja razširitev, ki so skupne za vse vrste razširitev, in specifične dobre prakse, ki so primerne za posamezne vrste razširitev. Tako si bomo ogledali, kako sami posežemo po kodi in s pomočjo programskega jezika PHP ustvarimo lasten gradnik. Za vsak tip gradnika je opisana natančno določena struktura nujno potrebnih datotek in map. Predstavljen je preprost vzorec, po katerem lahko uporabnik razvije lasten gradnik in ustvari namestitveno datoteko razširitve. V drugem delu razdelka so predstavljene značilnosti posameznih razširitev.

4.1 Splošne dobre prakse

Vse razširitve nujno potrebujejo le dve datoteki, in sicer datoteko php in datoteko templateDetails.xml. Vseeno je priporočljivo razširitve izdelati po programski arhitekturi MVC (angl. Model View Controller). Z uporabo MVC ločimo domensko logiko aplikacije od vnosa in predstavitve podatkov. Ta izboljšava omogoča veliko večjo fleksibilnost pri ažuriranju in večjo svobodo pri spreminjanju razširitve. Pri sistemih CMS spletne strani ne obstajajo fizično v obliki datotek HTML. Na zahtevo uporabnika, sistem CMS generira datoteke HTML, ki jih sestavi iz pravil za oblikovanje in podatkov v bazi in jih potem posreduje spletnemu brskalniku.

V tabeli 5 je predstavljena osnovna struktura datotek za predloge, komponente, module in vtičnike Joomla!. Večino datotek imamo lahko tudi le v eni mapi, vendar jih mnogi razvijalci raje smiselno organizirajo. Tako izboljšajo nadzor in pregled nad razširitvijo, lažje implementirajo spremembe in nadgradnje. Nujno potrebni datoteki za delovanje razširitev sta le XML in PHP, ki sta opisani v nadaljevanju.

Tabela 5: **Opis priporočene strukture map in datotek za razširitve Joomla!**

Priporočena struktura map in datotek za predloge Joomla!	Priporočena struktura map in datotek za komponente Joomla!
MOJA_PREDLOGA/ CSS/ index.php template.css SLIKE/ index.php mojapredloga.php templateDetails.xml	MOJA_KOMPONENTA/ SITE/ index.php mojakomponenta.php ADMIN/ index.php mojakomponenta.php SQL/ index.php UPDATES/ index.php MYSQL/ index.php 0.0.1.sql index.php MojaKomponentaDetails.xml
Priporočena struktura map in datotek za module Joomla!	Priporočena struktura map in datotek za vtičnike Joomla!
MOJ_MODUL/ TMPL/ index.php default.php index.php mojmodulDetails.xml mojmodul.php helper.php	MOJ_VTIČNIK/ mojvtičnikDetails.xml mojvtičnik.php

4.1.1 Datoteka XML

Datoteka XML (templateDetails.xml, MojaKomponentaDetails.xml, mojmodulDetails.xml in mojvtičnikDetails.xml v tabeli 1) je za razširitve bistvenega pomena, saj brez nje Joomla! ne prepozna razširitve.

Datoteka vsebuje informacije o razširitvi, opredeljuje datoteke, mape, določa položaje in konfiguracijske parametre za razširitev (Graf 2008).

V nadaljevanju je primer datoteke XML za razširitev tipa predloga:

```
<?xml version="1.0" encoding="utf-8" ?>
<!DOCTYPE install PUBLIC "-//Joomla! 1.7//DTD template 1.0//EN"
"flhttp://www.joomla.org/xml/dtd/1.7/template-install.dtd">
<extension version="1.7" type="template" client="site">
  <name>MojaPredloga</name>
  <version>1.7.0</version>
  <creationDate>15-12-2011</creationDate>
  <author>JK</author>
  <authorEmail>jaka.kuznik@gmail.com</authorEmail>
  <authorUrl>http://www.etus.si</authorUrl>
  <copyright>2011</copyright>
  <license>GNU/GPL</license>
  <description>Joomla! 1.7 predloga</description>
  <files>
    <filename>mojapredloga.php</filename>
    <filename>templateDetails.xml</filename>
    <filename>favicon.ico</filename>
    <folder>css</folder>
    <folder>slike</folder>
  </files>
  <positions>
    <position>top</position>
    <position>breadcrumb</position>
    <position>search</position>
    <position>right</position>
    <position>left</position>
    <position>footer</position>
  </positions>
</extension>
```

Podobno datoteko XML ima vsaka razširitev. Parameter type opredeli vrsto razširitve (language, template, component, module, plugin). Parameter client pa nam pove, za kateri del spletnega mesta je narejena razširitev (administrator, site, both). Pri oblikovanju nove razširitve je pomembno, da ima razširitev unikatno ime, ki se razlikuje od drugih razširitev. S parametrom method = "upgrade" v oznaki extension se vtičnik namesti, ne da bi odstranil starejšo različico. V tem primeru se prepisejo vse obstoječe datoteke, vendar stare datoteke ne bodo izbrisane. Parametri, ki navajajo podatke avtorja, datum, licenco in opis razširitve, niso obvezni.

4.1.2 Datoteka PHP

Datoteka PHP (mojapredloga.php, mojakomponenta.php, mojmodul.php in mojvtičnikDetails.php v ta-

beli 1) je glavna vstopna točka za razširitev. Opravlja vse potrebne vzpostavitevne rutine. V jedru datoteke PHP določimo funkcionalnost razširitve, vanjo lahko vključimo stilno predlogo in podporne datoteke. Pri komponenti je treba ustvariti datoteko PHP uporabniškega in skrbniškega dela strani. Datoteke PHP pa prav tako uporabimo za omejevanje dostopa do map, tako da v vsaki mapi izdelamo datoteko *index.php*, ki prikaže prazno stran. Te datoteke so lahko prazne ali vsebujejo preprosto kodo:

```
<html> <body bgcolor="#FFFFFF"> </ body> </ html>
```

4.1.3 Namestitvene datoteke

Množica imenikov in datotek ni primerna za distribucijo razširitev, zato v zadnjem koraku izdelave razširitev paket datotek skrcimo v eno datoteko. Izde-

lamo stisnjen arhiv, ki vsebuje imenik z zgoraj določeno strukturo (tabela 5). Paket je lahko v formatu ZIP (.zip), v obliki TAR-gzip (.tar.gz), ali v formatu TAR-bz2 (.tar.bz2). Tako pripravljeno datoteko namestimo preko skrbniškega dela strani Joomla! pod skupino funkcij razširitve. Po namestitvi je treba razširitev aktivirati oz. jo uporabiti na našem spletnem mestu. To storimo preko skrbniškega dela spletnega mesta z uporabo upravitelja razširitev Joomla! pravilno izdelano razširitev prepozna in jo glede na tip razširitve namesti na ustrezno lokacijo spletnega mesta.

4.2 Specifične dobre prakse

V nadaljevanju so prikazane specifične značilnosti posameznih vrst razširitev.

4.2.1 Predloga

Novejše predloge so narejene s plastmi (angl. layer), ki poenostavijo in zmanjšajo obseg oznak HTML, medtem ko večino izgleda opredelijo oznake v datoteki CSS. Takšen način je priporočljiv zaradi razumljivejše kode HTML, ki povzroči tudi boljše

uvrščanje strani – SEO. Starejše oz. preprostejše predloge imajo ogrodje izdelano s pomočjo tabel v HTML-ju, kjer je vstavljena tudi koda PHP za določanje položajev gradnikov Joomla!. Pri uporabi datoteke CSS moramo biti pozorni na možnost različne interpretacije oznak CSS s strani različnih brskalnikov.

V jedru datoteke PHP določimo položaje spletnega mesta, kamor bomo umeščali razširitve. V datoteko CSS zapišemo lego posameznih položajev, določimo barve in vse, kar je povezano z izgledom strani. Po namestitvi je treba predlogo nastaviti za privzeto in določiti vsebino, da se prikazuje v položajih predloge.

4.2.2 Komponente

Komponente delujejo tako na skrbniški kot tudi na uporabniški del spletnega mesta, zato je treba ustvariti dve glavni in enako poimenovani datoteki PHP (datoteki `mojakomponenta.php` za uporabniški in skrbniški del), ki se nahajata v različnih mapah. Ti določimo v datoteki XML, kot prikazuje spodnji primer:

```

<!-- datoteke uporabniškega dela strani -->
<files folder="site">
    <filename>index.php</filename>
    <filename>mojakomponenta.php</filename>
</files>
<!-- datoteke skrbniškega dela strani -->
<administration>
    <menu>Moja Komponenta</menu>
    <files folder="admin">
        <filename>index.php</filename>
        <filename>mojakomponenta.php</filename>
        <folder>sql</folder>
    </files>
</administration>

```

V datotekah PHP ustvarimo funkcionalnost komponente. Po namestitvi si lahko uporabniški del komponente ogledamo, tako da spletnemu naslovu spletnega mesta dodamo »`index.php?option=com_mojakomponenta`«. Administracija naše komponente je vidna iz skrbniškega menija komponente. Po namestitvi so dostopne funkcije, ki jih vsebujeta datoteki `mojakomponenta.php`, ki se nahajata v mapi »admin« in »site«.

4.2.3 Moduli

Modul lahko izdelamo na dva načina, in sicer z izdelavo modula v skrbniškem delu in z izdelavo modula z uporabo namestitvenih datotek.

Izdelava modula v skrbniškem delu je preprostejša, vendar je tak modul omejen na funkcije, ki so na voljo v skrbniškem delu. Prav tako ga moramo pri ponovni uporabi izdelati ponovno. Tak modul kreiramo kar v skrbniškem delu spletnega

mesta, izberemo gumb upravitelj modulov v skupini razširitve.

Drugi način je izdelava modula z namestitvenimi datotekami, pri čemer smo omejeni le z znanjem programskega jezika PHP. Poleg glavne datoteke PHP takšen modul običajno vsebuje še pomožne datoteke PHP, ki vsebuje razrede, ki se uporabljajo za zbiranje potrebnih podatkov. V glavni datoteki se sklicujemo na ustrezno metodo za pridobivanje podatkov, vključuje pa še predlogo za prikaz rezultatov. V bolj napredne module lahko vključujemo zahtevnejše podatkovne baze ali druge funkcije v tej datoteki. Razvoj modula je dokaj preprost, saj lahko z malo truda

razvijemo ponovno uporaben modul, ki popolnoma ustreza našim zahtevam.

4.2.4 Vtičniki

Vtičnik je zmogljiva razširitev, ki za razliko od modulov, komponent in predlog (delujejo v zgornji, razširitveni plasti sistema Joomla!) razširja funkcionalnosti v samem ogrodju sistema Joomla!. Vtičnike uporabljamo za prilagajanje osnovnih aplikacij, prav tako pa lahko izboljšajo uradne in neuradne razširitve. Osnova vtičnika je datoteka XML, v katero vključimo glavno datoteko PHP:

```
<files>
  <filename plugin="mojvticnik">mojvticnik.php</filename>
</files>
```

V datoteki PHP določimo učinek vtičnika. V nadaljevanju je prikazana koda preprostega vtičnika, ki bo spremenil naslov spletnega mesta. Joomla!

poimenuje določeno stran po naslovu vsebine, zato bomo naslovu dodali še ime spletnega mesta (Learn web design online).

Primer kode datoteke *mojvticnik.php*:

```
<?php
defined( '_JEXEC' ) or die( 'Restricted access' );

jimport( 'joomla.plugin.plugin' );

class plgSystemSiteNameInTitle extends JPlugin //Ustvarimo razred z JPlugin
{
function onAfterDispatch() { //funkcija upravlja dogodek onAfterDispatch
    global $mainframe;
    $document = &JFactory::getDocument();//dobimo dokument, ki ga potrebujemo
za dostop do naslova spletnega mesta
    $app = JFactory::getApplication();
    $document->setTitle($document->getTitle().' - '.$app-
>getCfg('sitename')); //nastavimo aeljen naslov spletnegamesta
    return true;
}
}
?>
```

Po namestitvi v »upravitelju vtičnikov« lahko omogočimo izdelani vtičnik. Tako se naslov spletne mesta spremeni iz »Domov« v »Domov – spletno mesto Joomla!«.

5 SKLEP

Sistem Joomla! zaseda pomembno mesto med sistemi za upravljanje vsebin. Popularnost mu omogočajo široka podpora, stalen razvoj in odprtokodna licenca GNU. Prednosti sistema Joomla! se vsak dan zaveda vse več uporabnikov, tako posamezniki kot tudi podjetja, ki z njegovo uporabo izkoriščajo konkurenčne prednosti učinkovitih odprtokodnih CMS-ov.

V prispevku smo predstavili sistem CMS, njegove prednosti in slabosti, s poudarkom na odprtokodnem sistemu Joomla!. Spoznali smo zasnovo, funkcionalnosti in razširitve omenjenega sistema. V tretjem razdelku smo predstavili pregled in analizo najpogostejših težav, na katere naletijo uporabniki sistema Joomla!, v četrtem pa smo na praktičnih primerih, podkrepjenih z izseki programskega koda, prikazali, kako priti do lastne predloge, modula, komponente in vtičnika. Z razširitvami postane sistem Joomla! še bolj zanimiv, saj ga lahko z njimi uporabniki prilagodijo in s tem maksimalno izkoristijo.

Predlagane rešitve pogostih problemov, ki so predstavljene v prispevku, so oblikovane tako, da olajšajo delo razvijalcem spletnih mest, ki sistem Joomla! šele spoznavajo. Prispevek jim bo pomagal hitreje prepoznati najpogostejše probleme sistema Joomla! in istočasno podati predloge, kako se jim izogniti. Zaradi pomembnosti področij zasebnosti, varnosti in zaupanja v spletna mesta smo v prispevku izpostavili vidik varnosti, ki vsebuje preproste napotke in rešitve za povečanje zaščite spletnega mesta. Predlagane rešitve so povezane med seboj, zato je pomembno, da jih obravnavamo kot celoto. Vzorčni modeli implementacije lastnih razširitev omogočajo uporabnikom večjo prilagodljivost spletnih strani. Pomembno sporočilo začetnikom sistema Joomla! je, da lahko s pomočjo poznavanja osnov programskega jezika PHP, v katerem je sistem Joomla! napisan, preprosto ustvarimo nove razširitve Joomla! oziroma prilagodimo obstoječe.

Težave sistema Joomla! se zaradi velike razširjenosti in velikega števila uporabnikov rešujejo dokaj hitro, prav tako pa se z izidi novih različic pojavljajo nove težave. S tem postaja področje vzdrževanja spletnih mest, ki temeljijo na CMS-ih, trajen proces.

Na področju razvoja lastnih razširitev je pomembno, da sledimo različicam sistema Joomla!, saj so razširitve kompatibilne le z različico, za katero so izdelane. Razlike med različicami so številne: v načinu namestitve, strukturi in v določenih funkcijah. Z nadaljnjimi raziskavami bi bilo treba slediti razvoju sistema Joomla!, opisati nove modele izdelave in postopke nadgradnje gradnikov za uporabo v naslednjih različicah.

6 LITERATURA

- [1] Bellamy, S. & Holzner, S.. (2011). Joomla! For Dummies. Indianapolis, IN: Wiley Publishing, Inc..
- [2] Center odprte kode Slovenije. (n. d.). Pridobljeno iz <http://www.coks.si>.
- [3] Creative Commons Slovenija. (n. d.). Pridobljeno iz <http://creativecommons.si>.
- [4] Creative Commons. (n. d.). Pridobljeno iz <http://creativecommons.org>.
- [5] Derr, M., Symes, T. (2008). Joomla!: Visual QuickStart Guide. Berkeley, CA: Peachpit Press.
- [6] Google trends. (n. d.). Pridobljeno iz <http://trends.google.com>.
- [7] Graf, H. (2008). Building Websites with Joomla! 1.5. PACKT.
- [8] Joomla Templates, Joomla Extensions for the Joomla CMS. (n.d.). Forum. Pridobljeno iz <http://designforjoomla.com/forum>.
- [9] Joomla! Forum. (n. d.). Pridobljeno iz <http://forum.joomla.org>.
- [10] Jowers, J. (2007). Open Source Pro: Joomla. Lulu press.
- [11] Juvančič, Boštjan in Juvančič, Primož. Birokrat. Ali smo sposobni sami skrbeti za spletne strani? 12 2006, Zv. 1.
- [12] Kramer, J. (2011). Joomla! 24-Hour Trainer. Indianapolis, IN: Wiley Publishing, Inc.
- [13] Kramer, J. (2010). Joomla! Start to Finish: How to Plan, Execute, and Maintain Your Web Site. Indianapolis, IN: Wiley Publishing, Inc.
- [14] Learn web design online. (n. d.). Pridobljeno iz <http://learnwebdesignonline.com>.
- [15] Marcofolio.net. (n. d.). Sedem nasvetov za izboljšanje varnosti v Joomla! Pridobljeno iz <http://www.marcofolio.net>.
- [16] Marriott, J. & Waring, E. (2010). The Official Joomla! Book. Boston, MA: Pearson Education.
- [17] Nasvet. (n. d.). Optimizacija spletnih strani. Pridobljeno <http://www.optimizacija-strani.si>.
- [18] North, Barrie M. (2009) Joomla! A user's Guide, Building a Successful Joomla! Powered Website. 2. izdaja. Boston, MA: Prentice Hall.
- [19] Open Source Matters. (n. d.). Pridobljeno iz <http://www.opensourcematters.org>.
- [20] Quinn, L. & Gardner-Madras, H. (2010). Comparing Open Source Content Management Systems: WordPress, Joomla, Drupal and Plone. Idealware.
- [21] Rahmel, D. (2009). Beginning Joomla! Second Edition. 2. izdaja. Berkeley, CA: Apress.
- [22] Rahmel, D. (2007). Beginning Joomla!: from novice to professional. Berkeley, CA: Apress.
- [23] SEO training. (n. d.). Pridobljeno iz <http://www.seotraining.org.uk>.
- [24] Shreves, R. (2009). Joomla! Bible. Indianapolis, IN: Wiley Publishing, Inc.

- [25] Skrt, R. (2006). Potreba po ažurnosti informacij. *Moj Mikro*, 2, 26–28.
- [26] Skrt, R. (2006). Prijubljeni in dostopni vsakomur. *Moj Mikro*, 2, 29–31.
- [27] Slovenska skupnost uporabnikov Joomla CMS. (n. d.). Pridobljeno iz <http://www.slojoomla.si>.
- [28] Snipt. (n. d.). Long-term memory for coders. Pridobljeno iz <http://snipt.net/nikosdion/the-master-htaccess>.
- [29] The GNU Operating System. (n. d.). Pridobljeno iz <http://www.gnu.org>.
- [30] Travis, B. (2011). *Pro Drupal 7 for Windows Developers*. New York, NY: Apress.

■

Jaka Kužnik je diplomiral na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru na področju računalništva in informatike. Zaposlen je v Iskri-PIO, d. o. o., Šentjernej. Pred tem je poslovne izkušnje nabiral v svojem podjetju. Med njegove interesne dejavnosti spadajo poglobljanje znanja programiranja za splet in učinkovite uporabe spletnih tehnologij.

■

Gregor Polančič je docent na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Med njegova interesna področja spadajo tehnološki in netehnološki vidiki sistemov za komuniciranje, sodelovanje, upravljanje vsebin in upravljanje informacijskih procesov vključno z njihovimi implikacijami.