

O KONCEPCIJI SIMPLE CONTENT MANAGERA (scm)

(Odnosi se na verziju scm_ver.0.2.1)

(1) Motivacija

Poticaj za pisanje programa scm (Simple Content Manager, prvotno SINBAD Site Code Maker) predstavljala je procjena njegovog autora da "klasični" CMS-ovi poput Joomla ili Wordpressa nisu adekvatni za posao slaganja web siteova u velikom broju slučajeva - zbog svoje veličine, kompliciranosti, resursa koje zahtijevaju, pa da za takav posao posve prikladan može biti i puno jednostavniji i manje zahtjevan softver. Nema sumnje kako je i inače koncepcija web programiranja i weba uopće koja se promovirala preko spomenutih "klasičnih" CMS-ova (uz određene koristi) donijela dosta štete, i što se tiče "tehničkih", i što se tiče "dizajnerskih" aspekata posla vezanog uz web, te da su zbog toga na ovom području potrebna neka nova rješenja. Premda određenih rješenja ovoga tipa ima, čini se da postoji neka vjerojatnost da se (jednog dana) pokaže kako scm (koji inače nije, i daleko je od toga da bude CMS u uobičajenom smislu te riječi, premda teži određenom dijelu njegove funkcionalnosti) ipak nije potpuno suvišan i da može relativno dobro poslužiti za slaganje jednostavnijih siteova. Inače, nadamo se da se kroz promociju ove vrste softvera može potaknuti i težnja prema jednostavnosti koje kod web dizajnera današnjeg vremena očito jako nedostaje.

(2) Općenito

Program scm napisan je u programskom jeziku C, a originalno je kompajliran pod operativnim sustavom Debian GNU/Linux 5.0 (Lenny), uz korištenje gcc_ver.4.3.2 kompajlera. Osim standardnih C-biblioteka scm uključuje i dvije LIn-ove biblioteke - libintpr i libpsutl. Program ima više opcija što se mogu zadati prilikom pokretanja (vidi opciju -h), a pritom je obavezno navesti naziv "instance" za koju se slažu .html dokumenti (web stranice). Termin "instanca" predstavlja zapravo skup konfiguracijskih fileova istog imena (= naziv_instance) u koje su spremljeni svi podaci potrebni za normalno generiranje predviđenih html dokumenata, s tim da je ovdje još potreban i inicijalizacijski (.ini) file za instancu čiji naziv ne mora nužno biti jednak nazivu instance (ako nije jednak potrebno ga je navesti kao argument programa pri pokretanju kreiranja html dokumenata - to je stavljeno tako da bi se s istom inicijalizacijom mogle kreirati različite varijante istog web sitea).

Podaci za inicijalizaciju koji pri radu programa vrijede generalno, spremljeni su u inicijalizacijskom fileu scm.ini.

(3) Instance fileovi

Svi podaci potrebni da bi scm mogao kreirati sadržaj web sitea odnosno predviđene html dokumente spremaju se u upravo spomenute konfiguracijske fileove, te u content fileove sa sadržajima specifičnim za pojedine web stranice. Konfiguracijskih fileova za danu instancu ima 7 - među kojima su 2 obavezna - osnovni (.conf) file i inicijalizacijski (.ini) file. Ovi fileovi su sljedeći:

naziv_instance.conf - osnovna konfiguracija za instancu

naziv_inicijalizacije.ini - inicijalizacija vezana uz instancu

naziv_instance.cfgi - dodatna inicijalizacija/konfiguracija za instancu

naziv_instance.cfgs - inicijalizacija/konfiguracija za sidebarove na stranicama

naziv_instance.cfad - liste content fileova koji se stavljaju u "repeating elemente" na stranicama

naziv_instance.imgs - popis slika koji se stavljaju na stranice

naziv_instance.lnks - popis linkova koji se stavljaju na stranice

Osnovna konfiguracija tj. file naziv_instance.conf sadrži definicije svih "page elemenata" i uključuje sadržaje koji je zajednički za sve stranice. Ostali fileovi sadrže razne druge podatke potrebne za kreiranje zadanih html dokumenata - primjerice .imgs i lnks fileovi omogućavaju "automatizaciju" postavljanja slika i linkova na stranice. Nazivi content fileova koji su pridruženi pojedinim html dokumentima koje treba generirati (pridruženi u smislu da obuhvaćaju sadržaj koji se stavlja u dokument tj. na web stranicu u procesu njezinog kreiranja) moraju biti oblika naziv_html_dokumenta.extens gdje extens predsta-

vlja tag tj. oznaku "page elementa" u osnovnoj konfiguraciji (.conf fileu) ili u content fileu na mjestu gdje se ubacuje odgovarajući sadržaj content filea - u sadržaje bilo kojeg content filea mogu se ubacivati drugi sadržaji tj. itemi označeni odgovarajućom labelom - iz istog content filea). Ako se isti sadržaji ponavljaju na više stranica, može ih se staviti u common content file čiji je naziv proizvoljan. Pravila za stavljanje sadržaja iz common content fileova navode se u inicijalizacijskom (.ini) fileu, gdje su također popisani svi nazivi html dokumenata koje treba kreirati, kao i extensa koji se koriste u konfiguraciji. Redak u ./conf fileu za instancu je oblika labela-sadržaj, dok je u content fileu oblika labela:sadržaj (ovi redci predstavljaju pojedine iteme tj. elemente sadržaja pridružene tagovima). Program uključuje i provjeru korektnosti config filea i svih content fileova koji se uključuju (ako redci nisu u navedenom obliku program se prekida). Content fileovi se inače mogu kreirati i automatski pomoću scm-a iz običnog (plain) teksta (opcija -f_). Pri pozivu ove opcije program automatski definira iteme (novi item predstavlja dio teksta odvojen s najmanje 3 retka od prethodnog), te stavlja line breakove tj.
 tagove (u slučaju kad je razmak manji), te bold tagove i ako je dio teksta okružen s dva ili više spaceova.

(4) Page elementi

Page elementi predstavljaju dijelove html dokumenta, odnosno koda koji su na odgovarajući način opisani u konfiguraciji, tako da se mogu jednoznačno "reproducirati" pri svome kreiranju tj. automatskom generiranju html koda. Postoje dvije vrste page elemenata - default page elementi i content page elementi. Default page elementi se definiraju u .conf fileu za instancu i njih čine obavezni basic elementi HEAD i BODY - koji odgovaraju dijelovima html dokumenta obuhvaćenim tagovima <head> i <body> i koji moraju biti navedeni (u .conf fileu) makar i "prazni" (bez sadržaja), te body elementi koji se ubacuju unutar elementa BODY. Body elementi su:

- PGHD - zaglavlje (header) stranice
- MENU - horizontalni menubar ispod zaglavlja
- BCNT - glavno "područje" stranice
- PGFT - podnožje (footer) stranice
- LSDB - lijevi sidebar
- RSDB - desni sidebar

Ova interpretacija odnosi se na neku "standardnu" formu stranice no prva četiri elementa zapravo je moguće interpretirati na proizvoljan način - jednostavno kao četiri predefinirana "horizontalna elementa", dok su ostala dva "vertikalni elementi" koji se mogu ubaciti u bilo koji predefinirani horizontalni element stranice, s tim da uvijek moraju ići oba zajedno (ne smije se ubacivati samo jedan), a ako se neki od njih ne koristi treba ga ostaviti "praznim" u .conf fileu (npr. treba staviti RSDB= ako se ne koristi desni sidebar).

Content elementi koji se definiraju u content fileovima uključuju specifične sadržaje za pojedine stranice, s tim da te sadržaje (kao i sadržaje default elementa) mogu činiti html tagovi i običan (plain) tekst. Osim ovih dvaju vrsta page elemenata postoje još i automatic elementi s oznakama IMG i LNK preko kojih se u html dokument automatski ubacuju slike i linkovi (definirani u spomenutim konfiguracijskim fileovima .lnks i .imgs). Napominjemo da je moguće dodati i nove default (body) elemente - dodavanjem novih njihovih naziva u file scm.h (preprocessorska varijabla vi). Pritom treba staviti i novi broj ovih elemenata (preproc varijabla vn) i dodati nove preproc varijable v_ (i u INIV_FIELD()), te naravno rekompajlirati program. Ako se dodaju novi vertikalni elementi treba promijeniti preproc varijablu AES koja označava njihov broj - sad će biti potrebno promijeniti i dijelove C-koda programa scm.c (vezane uz kontrolu slaganja ovih elementa na stranici). Nove horizontalne elemente treba navesti u preproc varijabli vi prije postojećih vertikalnih (LSDB/RSDB), a nove vertikalne iza njih (ovo je stavljeno tako jer za dodavanje novih vertikalnih elemenata sada nije potrebno mijenjati kod u scm.c).

Za page elemente postoji hijerarhija po kojoj oni mogu pozivati jedan drugoga - obavezni default page elementi mogu pozivati sve ostale, body page elementi mogu pozivati druge body page elemente kao i content elemente, a content elementi mogu pozivati samo druge content elemente, ali tako pozvani content tj. subcon-

content elementi mogu pozivati druge (sub)content elemente (ovdje treba paziti da pozvani elementi stoje "ispod" elemenata koji ih pozivaju u content fileu kao i da neki element ne poziva samog sebe (direktno ili indirektno) - premda program ne dopušta upadanje u beskonačnu petlju (pri čemu funkcija `create_scntpgelmnt()` stalno poziva sama sebe) u ovim slučajevima program javlja grešku. Postoje dakle četiri glavne "razine" u hijerarhiji pozivanja page elemenata, dvije razine za default elemente i dvije za content elemente (content i subcontent) s tim da na subcontent razini možemo elemente (tagove) pozivati rekurzivno do proizvoljnoga "stupnja". Default elementi mogu sadržavati i "repeating elemente" koji ne odgovaraju uobičajenim `|`-tagovima koji se koriste u scm-u već se "ograđuju" repeating oznakama (parom backslashova `\\`) - unutar ovoga "područja" ubacuju se sadržaji koji se uzimaju iz više content fileova i stavljaju u isti element jedan za drugim (o "ponavljanju" elemenata s includanim contentima više će riječi biti u sljedećem paragrafu).

Vežano uz page elemente u scm-u možemo reći da default elementi određuju generalnu formu web sitea, a content elementi formu pojedine stranice. Sadržaj stranica određen je uglavnom content elementima, premda ga djelomice određuju i default elementi (onaj koji je svim stranicama zajednički). Automatic elementi služe za pojednostavljivanje i ubrzavanje postupka slaganja konfiguracije (za danu instancu).

(5) vbar tagovi (`|`-tagovi) i repeating oznake backslash-backslash (`\\`)

vbar ili `|`-tagovi predstavljaju oznake za page elemente tj. opisuju pravilo po kojemu se u određeni page element ubacuju sadržaji iz drugih elemenata odnosno fileova (tako ih zovemo zato jer su smješteni između dva "vbar" znaka) Ovi tagovi mogu biti oblika:

`|default_page_element|` - default tagovi - mogu stajati samo u osnovnoj konfiguraciji (`.conf`) fileu za instancu, i to jedino u elementima koji su na istoj ili nižoj razini u "hijerarhiji" (ne mogu pozivati ni sami sebe).

`|extens$labela|` - content tagovi - mogu stajati samo u `.conf` fileu instance - i to u bilo kojem elementu (extens je ekstenzija content filea iz kojega se uzima potrebni sadržaj, a labela odgovara itemu toga filea čiji se sadržaj ubacuje na mjesto ovoga taga. Različiti tagovi ove vrste unutar istog page elementa moraju umati isti extens (pri njihovoj interpretaciji čita se samo jedan content file).

`|labela|` - subcontent tagovi - mogu stajati samo u content fileovima i preko njih se mogu pozivati samo sadržaji (itemi) iz istog content filea koji u sebi ne sadrže druge tagove. Ako u istom content fileu ima više itema s istom tom labelom, svi se oni jedan za drugim stavljaju na mjesto odakle su pozvani - unutar taga može se staviti i "okruženje" pozvanog itema - čiji su elementi ograničeni delimiterom (`;`) (primjerice `<center>;labela;</center>;|`).

Uz to postoje "tagovi za automatizaciju" oblika:

`|IMG$labela|` - oznaka za ubacivanje slike gdje labelu predstavlja broj (simbol) odgovarajuće slike u `.imgs` fileu

`|LNK$labela|` - oznaka za ubacivanje linka gdje labelu predstavlja broj (simbol) odgovarajućeg linka u `.lnks` fileu

Ovi tagovi se mogu ubacivati u bilo koje druge page elemente no sami ih ne mogu pozivati - jedino se unutar links taga može pozvati images tag (samo jedan).

U slučaju da se sadržaji content filea stavljaju u tablice, mogu se koristiti i oznake (tagovi) za ubacivanje odgovarajućih "table tagova" predstavljeni znakom `"^"` (ASCII 94) - pravila za ove supstitucije definiraju se unutar content filea (korištenjem "CTT-simbola").

Repeating oznake (`\\`) predstavljaju delimitere html koda sa content tagovima za koje se sadržaj uzima iz fileova definiranih u odgovarajućem retku `.cfad` filea za instancu - taj redak počinje sa naziv_html_stranice.extens (radi se o stranici koja se upravo kreira i ekstensu iz upravo spomenutog content taga). Ovi se sadržaji ubacuju jedan za drugim - po redu kojim su content fileovi navedeni u `.cfad` fileu i njih može biti proizvoljno mnogo, osim ako taj broj (radi sigurnosti) nije ograničen odgovarajućom varijablom iz `scm.ini`. Inače, scm sadrži opciju (`-a`) kojom se automatski dodaje sadržaj jednog tekstualnog filea na određenu stranicu (tekstualni file se pritom i formatira tj. pretvara u content file te se dodaje u odgovarajući redak `.cfad` filea - tako da se ovaj sadržaj uzima u obzir i pri ponovnom kreiranju cijeloga sitea za danu instancu). Unutar repeat-

ing oznaka ne smiju se ubacivati default tagovi. Stranice koje nisu navedene u .cfad fileu kreiraju se kao da repeatinga nema.

Napominjemo da je unutar repeating elementa moguće zadati poseban (html) kod za stranice na kojima nema repeatinga - za to je potrebno na kraj "repeating" koda tj. sadržaja staviti oznaku "^" (ASCII 94) i iza te oznake staviti nonrepeating kod tj. sadržaj za odgovarajući page element. Prema tome sve stranice na kojima nema repeatinga možemo kreirati prema drugim pravilima u odnosu na one s repeatingom (s tim da znak ^ predstavlja delimiter repeating i nonrepeating sadržaja unutar \\ oznaka).

(6) Kako stvar radi?

scm radi tako da čita konfiguraciju i slaže html kod za sve web stranice koje su zadane u inicijalizacijskom fileu. Prvo se učitavaju podaci za inicijalizaciju - nazivi html stranica koje treba kreirati, nazivi ekstensa, nazivi common content fileova i pravila kojim oni supstituiraju odgovarajuće kombinacije naziva stranica i naziva ekstensa (ovi nazivi označavaju se rednim brojem koji predstavlja njihovu poziciju u (;)-delimited nizu ovih naziva u .ini fileu, pa se radi o kombinacijama tih brojeva, s tim da brojanje počinje od 1). Zatim se učitavaju .imgs i .lnks fileovi odnosno podaci o linkovima i slikama, a potom otvara .conf file za instancu u kojemu se najprije čitaju bazični default elementi HEAD i BODY (<head> i <body> tagovi stavljaju se automatski) i nakon toga se iz njih, po potrebi poziva sve ostalo - content fileovi otvaraju se tek kad program naiđe na odgovarajući content tag u konfiguraciji - pritom se učitavaju svi itemi iz content filea, a ne samo onaj potrebnii tako da su odmah na raspolaganju (to je inače razlog zašto ekstensi unutar istog page elementa moraju biti isti, što je već spomenuto ranije). Ovaj postupak se ponavlja za sve html stranice koje se kreiraju pri obradi. U slučaju greške pri postupku kreiranja stranice, stranica se složi djelomično (dokle ide), i na mjestu prekida se stavi se obavijest da stranica nije kompletna. O svim greškama koje nastaju pri obradi program izbacuje odgovarajuću poruku na promptu. Program inače sadrži i opciju (-e) za provjeru da li postoje svi fileovi navedeni u inicijalizaciji - trebalo bi još dodati provjeru postojanja svih slika i dokumenata navedenih u .imgs i .lnks fileovima, te provjeru da li možda .conf file sadrži i ekstense tj. labele koje ne postoje. Program sadrži i opciju (-d) za kreiranje html dokumenata u zadanom direktoriju. Kompletna konfiguracija osim eventualno .ini filea mora se nalaziti u istom direktoriju - s tim da je ovaj direktorij određen argumentom programa koji označava naziv instance.

(7) Osnovni koraci pri kreiranju html stranica za web site

- (1) Pripremiti sadržaje svih stranica u .txt (plain tekst) formatu i urediti ih (po potrebi) prema opisanim pravilima za formatiranje content fileova
- (2) Formatirati ove tekstove pomoću scm-a (opcija -f)
- (3) Popisati sve slike i linkove u .lnks i .imgs fileu (u skladu sa formatom ovih fileova) - taj se posao može djelomice automatizirati pomoću priloženih bash skripti - cimgfile.sh i clikfile.sh
- (4) Kreirati konfiguraciju za danu instancu (složiti konfiguracijske fileove .conf, .ini, .cfgi, .cfgs, .cfad) i po potrebi prilagoditi globalni .ini file (scm.ini)
- (5) Dodatno urediti content fileove - po potrebi
- (6) Kreirati site (sve html stranice) pokretanjem scm-a:
~/scm [opcije] ime_instance [ime_inicijalizacije]

(131214/140104)