Tartalomkezelés

Szereljük fel honlapunkat hírújságszintű tartalomkezeléssel – nyílt forrású program használatával.

Emlékszünk még a Web régi szép napjaira, amikor a webmester volt a mindenes, egymaga csinált mindent a grafikus tervezéstől kezdve az adatbázis-programozáson át egészen a DNS-táblamódosításokig? Ahogy a Web fejlődött, az egyemberes weblapok egyre ritkultak. Igaz ugyan, hogy még ma is viszonylag egyszerű dolog egy szál magunkban egy dísztelen honlapot létrehoznunk és karbantartanunk, de általában már a legkisebb szervezetek is megosztják a felelősséget a programozók, a tervezők és a tartalmat adó emberek között. Továbbá számos szervezet kifejezetten azt szeretné, ha a különböző tartalomtípusokért különböző emberek felelnének, miközben valamennyien teljes jogkörrel rendelkeznének egy adott rész felett.

Természetesen a kiadók világában ez nem számít újdonságnak. Még a gimnáziumi hírújságot szerkesztettem, de már akkor használtuk az Atex nevű szerkesztő- és szedőrendszert. Az Atexet több dologért is kedvelték, de leginkább azért, mert épp úgy működött, mint az újság. Az Atexet használó riporterek egy súlyos, egyedi billentyűzeten a ***küld*** gomb megnyomásával elküldték írásaikat a szerkesztőknek. A szerkesztők végignézhették, hogy milyen cikkek várakoznak elolvasásra, szerkeszthették is őket, sőt a cikket vissza is küldhették írójának, vagy továbbíthatták a szövegszedő részleghez. A felépítésből fakadóan tilos volt olyan cikkeket megnézni, módosítani vagy fogadni, amiket a feldolgozási folyamat következő résztvevőjének továbbküldtek. A „Torkig vagyok a sürgetéssel!" kiáltással felpattanó riporter képe romantikus és lelkesítő lehet, de valószerűtlen a mai világban, ahol a híripar igen szűk határidőkkel dolgozik.

Ahogy a weblapok elérik az újságok szintjét, nem szabad meglepődnünk azon, hogy olyan programokat kezdenek használni, amelyeket tartalomkezelő rendszereknek (Content Management System, azaz CMS) nevezünk, és ezek igen hasonlóan működnek a korábbi Atexhez. A dokumentumok szervezése, az emberek és a munkafolyamat irányítása azonban összetett feladat, különösen, ha minden ember igényeit egyetlen programcsomagba szeretnénk belepakolni. Így aztán annak ellenére, hogy a tartalomkezelés egyre több weblap számára létfontosságú, a CMS-ügynökök híresek arról, hogy feltupírozott, drága programokat adnak el, amelyek sokat ígérnek ugyan, de vajmi keveset nyújtanak.

Mit csinál a CMS?

A tartalomkezelés egyik nehézsége, hogy minden weblap igényei mások. Ezért aztán az üzleti CMS-programokat általában két részletben árusítják. A vásárló először az alapprogramért fizet, majd legalább még egyszer ugyanennyit kell fordítania a tanácsadó és támogatási szolgáltatásokra. Így a CMS-program nemcsak drága, de igen jelentős megvalósítási és kipróbálási időt is igényel. Más szavakkal a CMS általában közelebb áll az eszközkészlet fogalmához, mint a kész alkalmazáséhoz. A legtöbb ilyen eszközkészlet a következő elemeket tartalmazza:

- ***Felhasználók:*** ha a weblapon mindenkinek különböző jogokat szeretnénk biztosítani, nyilván minden felhasználónak külön bejelentkezőnevet kell adnunk. A CMS ennek megfelelően tartalmaz egy felhasználókezelő programot, ami elkészíti, törli, szerkeszti és letiltja a rendszer felhasználóit.
- ***Engedélyek:*** miként Linux alatt képesek vagyunk írási, olvasási és végrehajtási jogosultságokat kiosztani a különféle fájlokra vonatkozóan, általában a CMS-program is lehetővé teszi, hogy az oldal karbantartója különféle jogokat adjon a rendszer összes felhasználójának. Hírújsághasonlatunkhoz visszatérve: a riporterek bevihetik a tartalmat, a szerkesztők módosíthatják (vagy visszaküldhetik a riporternek), végül a kiadók nyilvánosan elérhetővé tehetik (vagy visszaküldhetik a szerkesztőnek).
- ***Csoportok:*** bár elméletileg az egyes felhasználókhoz is rendelhetünk jogokat, ez a módszer hamar fárasztóvá válhat. Ezért a legtöbb CMS-program lehetővé teszi, hogy a felhasználókat az engedélyek kiosztása érdekében csoportosítsuk. Például megadhatjuk, hogy Tom, Dick és Harry írhatnak, szerkeszthetnek, de nem publikálhatnak, vagy ezeket a jogosultságokat hozzárendelhetjük az ***általános nevek*** csoporthoz – a hatás ugyanaz lesz.
- ***Sablonok:*** számos sablonozórendszer létezik, ilyen a JSP, HTML::Mason és a PHP is. A legjobbak szétválasztják egymástól a tervezést és a programlogikát, így a tervezők, írók és programozók anélkül tudnak egyszerre dolgozni a lapon, hogy egymás tyúkszemére lépnének.
- ***Kiadás:*** a Web kétélű fegyvere az azonnaliság. Amint módosítjuk kiszolgálónkon a ***foo.html*** lapot, mindenki láthatja, min változtattunk. Mi történik, ha hibázunk? Mi lesz, ha előbb ki szeretnénk próbálni? A CMS megoldása erre az, hogy minden tartalomdarabot kiadottnak kell megjelölni, mielőtt az láthatóvá válna a nagyközönség számára. Amíg a cikk nincs kiadva, addig láthatatlan.
- ***Előzetes és előnézet:*** miképpen a hírújságok és magazinok kiadói is szeretnék látni, hogyan fog kinézni a végleges termék, mielőtt elkezdenék nyomtatni a tényleges másolatokat, azonképpen a webkiadók is meg szeretnék tekinteni az oldalt, mielőtt az élővé válna a Weben. Ezért sok oldal futtat előzetes nézetet tartalmazó kiszolgálót, ami a legtöbb szempontból azonos a kiadási kiszolgálójukkal, attól eltekintve, hogy a világ elől el van rejtve. A kipróbálás ezeken az előnézeti kiszolgálókon folyik; amikor a szerkesztő vagy a kiadó elégedett, a tartalom átkerül a kiadási kiszolgálóra. Egy igazi CMS szinte bizonyosan lehetővé teszi, hogy ilyen formában állítsuk fel a rendszerünket.
- ***Munkafolyamat:*** az előnézet készítése már a végső lépés, amit igen hosszú utazás előz meg a szerző munkaállomásától a kiadási kiszolgálóig. Azt a módot, ahogyan a tartalom végigmegy a rendszeren, munkafolyamatnak (work flow) nevezzük. A CMS feladata legnagyobbrészt abból áll, hogy

ezt a munkafolyamatot meghatározza és kezelje. A riporterek visszaránthassák a történeteket a szerkesztőiktől? Hány szintű szerkesztői rendszert szeretnénk? Hová kerülnek a tervezők? Ki adja ki a végső kiküldési utasítást a tartalomra? Az összes ilyen kérdést a CMS munkafolyamat része kezeli.

- ***Kiadási dátumok:*** a Web előnye, hogy a kiadott anyagok azonnal megjelennek. De mi a helyzet, ha a vállalatunk részvénymanipulációt szeretne bejelenteni, de ezt az adatot nem teheti közzé hétfő kilenc óra előtt? Ülhetünk a számítógép mellett, várva, hogy a mutató elérje a kilenc órát, majd lenyomhatjuk az ENTER billentyűt, végre mindenki előtt felfedve a dokumentumot. Avagy használhatunk CMS-t, ami általában lehetővé teszi, hogy megadjuk, mikor szeretnénk megjeleníteni a cikket, illetve mikor fog lejárni.
- ***Webalapú szerkesztés:*** bár komoly szövegszerkesztésre a webböngésző az egyik legrosszabb választás, a legtöbb CMS-rendszer lehetővé teszi, hogy bizonyos vagy az összes dokumentumot böngészővel szerkesszük. Igazság szerint szinte minden CMS lehetővé teszi azt is, hogy fájlokat töltsünk fel a helyi számítógépünkről. A webalapú szerkesztés olyankor lehet hasznos, amikor sietünk vagy éppen csak egy-két dolgot szeretnénk megváltoztatni. Természetesen amelyik CMS lehetővé tesz ilyesfajta szerkesztést, az azt is ellenőrzi, hogy a lapot szerkeszteni próbáló személy rendelkezik-e a megfelelő jogosultságokkal.
- ***Keresés:*** a legtöbb CMS-csomag valamilyen keresési lehetőséget is nyújt, így dokumentumainkat megtalálhatjuk a rendszeren.

Bár fenti listánk nem teljes, viszonylag átfogó képet ad arról, hogy milyen nehézségeket próbál a CMS megoldani. Mindenki sejtheti, hogy minden CMS egy kicsit más képességkészletet tartalmaz és más módszert alkalmaz e nehézségek legyűrésére. Minthogy a CMS az ideje nagy részét tárolással, lekéréssel és tartalomkövetéssel tölti, senkit sem érhet meglepetésként, hogy a CMS számára az adatbázis majdhogynem létfontosságú.
Az üzleti CMS-csomagok általában azt feltételezik, hogy üzleti adatbázisrendszert használunk, például Oracle-t vagy a Microsoft SQL Serverét. Ahogy az várható, a nyílt forráskódú CMS-programok általában úgy készülnek, hogy a legjobban az olyan nyílt forrású adatbáziskezelőkkel működjenek, mint a MySQL vagy a PostgreSQL. A Zope tartalomkezelő keretrendszere (Content Management Framework, avagy CMF), ami tulajdonképpen egy saját CMS készítéséhez szánt eszközkészlet, szintén adatbázist használ, de ebben az esetben az adatbázis külső relációs adatbázis helyett a Zope beépített objektum-adatbázisa (ZODB) lesz.

Tartalomkezelés kontra alkalmazásfejlesztés

Ha valaha már fejlesztettünk komoly webalkalmazásokat, valószínűleg azonnal feltűnik a CMS által felajánlott szolgáltatások és web-alkalmazáskiszolgálótól elvárt képességek közötti hatalmas átfedés. A legtöbb CMS-program egy web-alkalmazáskiszolgáló felett helyezkedik el, s annak lehetőségeit használja fel a HTTP-kapcsolatok, a felhasználók, a csoportok, a jogok és az adatbázis-API kezeléséhez. Tulajdonképpen a CMS volt a weben fejlesztett első népszerű alkalmazásosztály, mint ahogyan a szövegszerkesztők voltak az első alkalmazások a személyi számítógépeken.
Általában véve nem rossz dolog, hogy a CMS-program az alkalmazáskiszolgálóra épül, és különösen igaz ez a nyílt forráskód világában. Így ugyanis új modulokat adhatunk a CMS-maghoz, új dokumentumtípusokat kezelhetünk, megváltoztathatjuk a sablonokat, kibővíthetjük az adatbázist, új jogtípusokat és munkafolyamat-szabályokat adhatunk meg. Nem szabad azonban megfeledkezni az alkalmazáskiszolgáló és CMS közötti különbségekről sem. Az első hátteret biztosít az alkalmazások kifejlesztéséhez, a második egy olyan alkalmazás, amit testreszabhatunk.
Amennyiben webalapú újságot, magazint vagy vállalati hírlapot szeretnénk létrehozni, nem kétséges, a CMS az, amire szükségünk van. Ellenben ha olyan webalapú alkalmazást szeretnénk, ami kedvenc jótékonysági intézményünk adományait kezeli, könnyen lehet, hogy a CMS-ben nem találjuk meg a szükséges rugalmasságot. A webalkalmazás és a webkiadás közötti különbség mindig is egy kicsit homályos volt, de ahogy a webalkalmazások kifinomultabbakká válnak, egyre inkább látható, hogy a CMS-program egy termék, amit webes felületen futtathatunk.
Mivel tartalomkezelőnk általában valamilyen alkalmazáskiszolgáló felett fog futni, a kiválasztott CMS nagyban függhet az általunk használt kiszolgálótípustól. Számos cég váltott át J2EE (Java 2 Enterprise Edition) alapra. Tulajdonképpen a jól ismert Vignette CMS-t is eredetileg Tcl nyelv alapúra tervezték, majd később átvitték J2EE alá, amikor a J2EE körüli láz olyan nagyra nőtt, hogy már nem lehetett figyelmen kívül hagyni. Mivel J2EE inkább szabvány, semmint termék, a vásárlók külön kiválaszthatják az alkalmazáskiszolgálójukat és CMS-programukat. Használhatjuk a Tomcat-JBoss párost vagy a BEA, az IBM és hasonló cégek üzleti ajánlatait.
Amennyiben nem kedveljük a Javát, vagy a fejlesztőcsapatunk más technológiákban jártasabb, kereshetünk nem J2EE-s CMS-t is. Ilyen termékek is léteznek, néhányukat meg is nézzük a következő hónapok során. Ilyen rendszer a Zope CMF-e, a CMF alapú Plone, Bricolage (Perl-PostgreSQL), PHPNuke-PostNuke/Xoops (PHP) és a Midgard (PHP).
A weblapkiadás területén a CMS használata egyre inkább szükségszerű lesz. Még akkor is érdemes lehet CMS-re váltanunk, ha csak egymagunk dolgozunk a weblapunkon, hiszen a honlapunkon segít szabványosítani a kinézetet és a tartalomtovábbítást.

Linux Journal 2003. április, 108. szám

Reuven M. Lerner (➲ http:/www.lerner.co.il/atf) Nyílt forrású programokra, valamint web- és adatbázis-alkalmazásokra szakosodott tanácsadó. Könyve, a Core Perl, 2002 januárjában jelent meg a Prentice Hall gondozásában. Reuven feleségével, és lányával Izraelben, Modi'in-ben él.

KAPCSOLÓDÓ CÍMEK

Az elfogulatlan CMS-adatok legjobb forrása a CMS-levelezőlista ➲ http://www.cms-list.org. A lista jelenlegi archívuma e cikk írásakor nem érhető el – de mint mondtam, a lista felbecsülhetetlen értékű áttekintést ad a CMS-piacról, üzleti és nyílt forrású termékek terén egyaránt.

A CMSWatch weblap (➲ http://www.cmswatch.com) vegyesen tartalmaz részletes jelentéseket, ingyenes tanácsadást, összefoglalókat és bemutató írásokat. Hasznos oldal, ha CMS-programokkal kapcsolatos anyagokra vadászunk.