

Dinamikus alkalmazásfejlesztés

Az eszközválasztás szempontjai.

Nemrég abban a megtiszteltetésben részesültem, hogy egy nagy, philadelphiai orvosi kiadó cég internetes részlegénél dolgozhattam Pennsylvániában. A részlegünk a cég számos marketingosztályának nyújtott segítséget különféle Internetre szánt anyagok kezelésében, illetve irányította és ütemezte a tartalom elhelyezését a honlapján. A részlegünk gyakorlatilag összekötőként szolgált a kiadó különféle marketing- és termékfelügyeleti osztályai között Philadelphióban, New Yorkban, Chicagóban és a Baltimore-i IT részlegben, ahol a webkiszolgálók üzemeltek.

A tevékenységgel együtt járt, hogy hatalmas mennyiségű anyag került a legkülönfélébb utakon a részlegünköz. A legtöbb adat elektronikus levélben érkezett, egyes fájlok lemezen jöttek az irodák közötti postával, míg továbbiakat közvetlenül a Baltimore-i IT részlegen található ftp-kiszolgálókra töltöttek fel. A kavarodást csak tovább növelte, hogy az anyagokból sokszor többféle változat is létezett, mivel a részlegek még az utolsó pillanatban is módosítottak rajtuk. A feladatom az volt, hogy a különböző tartalmakat központi alkalmazással szervezve ideiglenes megoldást dolgozzak ki a különféle anyagok érkezése miatt keletkező kavarodás mérséklésére. Ezt egy kisebb webes alkalmazással igyekeztem elérni, mely az egyes tervezetek címét, kezelőjét, részlegét, részlegének kódját és költségelszámolási kódját volt hivatott követni.

Az alkalmazást gyorsan és olcsón kellett elkészíteni, mivel a vásárlások visszafogása és bizonyos beszerzési folyamatok miatt a következő pénzügyi évig nem volt mód jelentősebb fejlesztésre. Teljesen hétköznapi PC-s felszerelést kaptam, a részleg már meglévő eszközeiből. Az alkalmazásnak kicsinek, gyorsnak, könnyen használhatónak kellett lennie, további követelmény volt, hogy a részleg személyzete is képes legyen karban tartani. Ők viszont inkább üzleti, és nem szakmai szempontból kezelték a világhálót. Annak ellenére, hogy ez volt az internetes részleg, tevékenysége inkább a tartalom felügyeletére korlátozódott. Az IT részleg kezelte a nagyobb terhelésű webkiszolgálókat és az irodán kívüli távközlési eszközöket.

Célkitűzés

Írásom egyszerűbb dinamikus internetes alkalmazás készítéséhez szükséges és elérhető eszközök tanulmányozása során szerzett tapasztalataimra épül. Az összefoglalást a kereskedelmi és ingyenes programokhoz egyaránt elkészítettem. Néhány meghatározással fogjuk kezdeni, illetve a dinamikus internetes alkalmazások háttérrel, majd áttekintünk néhány elérhető módszert. Szándékom szerint a cikk kiindulópontként szolgál majd a jelenlegi és leendő fejlesztők számára, ha internetes alkalmazásaik fejlesztéséhez eszközöket keresnek.

Milyen egy dinamikus honlap?

A haladó honlaptervezők a HTML és a CSS mellett gyakran egy JavaScriptnek nevezett parancsnyelvet, illetve a honlap összetevőinek elnevezési rendszerét – a Dokumentum Objektum Modellt, avagy a DOM-ot – használják dinamikus tartalom készítésére. Az eredményt gyakran dinamikus HTML-nek, vagy DHTML-nek is nevezzük. A megvalósítás során készített parancsfájlok a JavaScripttel ellentétben a kiszolgálón, és nem az ügyfélnél futnak. Az itt ismertetett eszközök és módszerek lehetővé teszik, hogy a későbbiek folyamán további interaktív vagy stílusbeli elemekkel bővítsük az alkalmazást.

Mi az adatbázis?

Az adatot tények, számok, betűk és jelek összességéként határozzuk meg, amit számítógép vagy távközlési rendszer dolgoz fel, hogy értelmezhető adatot állítson elő belőle. Számítógépes rendszerekben ezeket az elemeket jellemzően állományokban tárolják. Az egymással összefüggő állományok adatbázist alkotnak. A fájlokban belül az elemeket sorokba és oszlopokba rendezzük. Ebben az esetben azonban valami kifinomultabb rendszerre van szükség, mint állományok halmazára.

A tervezethez egy relációs adatbázis-kezelő rendszer (RDBMS) szükséges. Az RDBMS olyan programcsomag, mely az adatokat sorokból és oszlopokból álló táblákban tárolja. A különböző táblák közt relációk hozhatók létre, így adható válasz a felhasználó által feltett kérdésekre. Ezeket a kérdéseket lekérdezéseknek nevezzük. Az RDBMS elgondolást először *Codd* vezette be 1970-ben egyetemi tanulmányában, kereskedelmi forgalomban azonban a hetvenes évek közepéig nem létezett ilyen rendszer. Az RDBMS választ ad minden lekérdezésre, függetlenül attól, hogy az adatokat lekérdezzük, frissítjük vagy töröljük a táblából.

A honlapot kiszolgáló RDBMS-t gyakran háttérrendszernek is nevezük. A honlapokat, melyeket a felhasználó lát, megjelenő szolgáltatásnak hívjuk. A háttérrendszernek az SQL (Structured Query Language – rendezett lekérdezőnyelv) használatával tehető fel a kérdések. Röviden összefoglalva, ha adatokat kell feldolgozni, valamilyen DBMS-re mindenképpen szükség lesz. A legnépszerűbb rendszerek napjainkban a relációs DBMS-ek, melyek SQL lekérdezésekre is parancsokra adnak választ.

Az adatbázis kiválasztása

Oracle

Az Oracle olyan termék, melynek neve az adatbázis szó rokon értelmű kifejezésévé vált. Az Oracle (☞ <http://www.oracle.com/>) jelentős szerepet játszott abban, hogy jelenleg igen elterjedtek a relációs adatbázisok. Az évek során adatbázis-kiszolgálója jelentős elismerést vívott ki, mint teljes szolgáltatáskészletet nyújtó, gyors és megbízható rendszer. Az Oracle a Linuxot is támogatja, és a jelek szerint a linuxos rendszerek elkötelezettje.

Ennek ellenére van két jelentős ok, amiért ennél a fejlesztésnél nem használtam Oracle-t. Először is a gépre vonatkozó követelményei bőven meghaladták az alkalmazás fejlesztéséhez és kiszolgálásához rendelkezésre álló gépét. Mindjárt kezdetként 800 MB helyre van szükség a merevlemezen, és 256 MB memóriára. Másodszor az Oracle túlságosan drága választás lett volna ekkora „átmeneti” alkalmazáshoz. Még a legkevesebb szerződési költséget véve is, öt át nem ruházható felhasználói joggal egyetlen kiszolgálóra is 800 dollárt kellett volna fizetni a program használatáért, azaz felhasználónként 160 dollárt.

MySQL

A felhasználók számos nyílt forráskódú alkalmazást érhetnek el. Figyelembe véve a működő honlapokat, a MySQL (☞ <http://www.mysql.org/>) rendkívül népszerű választásnak tűnik a Linux-közösségen belül. A MySQL nagyon gyors, többszálú, többfelhasználós és erőteljes SQL adatbázis-kiszolgáló. A MySQL jelenleg szintén nyílt forráskódú, és nemrég szövetségre lépett a VA Linux Systems céggel, mely linuxos

Kapcsolódó címek

Axmark, 2000 MySQL Manual – a kézikönyvet *David Axmark, Michael (Monty)*

Widenius és Paul DuBois készítette és gondozza

➔ <http://www.mysql.com/documentation/mysql/commented/>

Allaire ➔ <http://www.allaire.com/>

Gilmore, 2000

➔ http://www.oreillynet.com/pub/a/network/2000/06/16/magazine/php_mysql.html

MySQL Org ➔ <http://www.mysql.org/>

Oracle, 2000, Press Release

➔ <http://www.oracle.com/corporate/press/index.html?134934.html>

Oracle FAQ, 2000, Oracle 8i on Red Hat 6.1 Installation FAQ

➔ <http://platforms.oracle.com/linux/>

Oracle árak, 2000 ➔ <http://oraclestore.oracle.com/>

Perl Org ➔ <http://www.perl.org/>

PHP ➔ <http://www.php.net/>

PostgreSQL ➔ <http://www.postgresql.org/>

PostgreSQL Admin Guide, melyet Thomas Lockhart szerkeszt

➔ <http://www.postgresql.org/doxlist.html>

PostgreSQL FAQ, 2000

➔ <http://www.postgresql.org/docs/faq-english.html>

gépeket árusít és támogat. A MySQL első nyilvános kiadása 1996 novemberében jelent meg, és kezdettől fogva a forráskóddal együtt volt elérhető. A MySQL bizonyítottan villámgyors és megbízható adatbázis-megoldás, melyet egyre több cég használ. Többek közt az SGI, a ValueClick, a Nortel/Insight, a Tucows.com, a Cisco és sok más cég bizalmát is elnyerte.

Úgy tűnt, hogy a MySQL több mint megfelelő megoldás a kezdetben leírt egyszerű kis alkalmazáshoz. Nagy méretű rekordhalmazokkal is rendkívül gyorsnak bizonyul. A MySQL kézikönyve ötvenmilliónál több rekordot tartalmazó rendszerekről is beszámol. Ezenkívül egyre szorosabb együttműködés jelei mutatkoznak a MySQL és a PHP fejlesztői csapatok között. A dinamikus páros növekvő népszerűsége, társulva mindkét oldal fő fejlesztőinek határtalan lelkesedésével idén az izraeli találkozóban csúcsosodott ki. Az eredmény a PHP 4.0 csomaggal együtt közreadott MySQL könyvtár lett, illetve egy meg-egyezés, miszerint a felek kölcsönösen segítik egymást a termékek együttműködésének és minőségének javításában.

Ennek ellenére a MySQL használatának is van néhány hátránya. Ezek egyike a tranzakciók területén jelentkeznek. A tranzakció (néhány nyelvjárásban: ügylet) egymással összefüggő változtatások halmaza egy adatbázisban. Az SQL szabvány előírja, hogy változtatások egész sora adható ki az adatbázisnak, majd oszthatatlan egységként hajtható végre vagy vonható vissza. Így lehetővé válik, hogy például az adatbázis különböző tábláiban található számlaszámok között pénzt mozgassunk át úgy, hogy az egyik számlához hozzáadjuk az összeget, majd megpróbáljuk levonni a másiktól. Ha a másik változtatás sikertelen, az egész műveletsort egyetlen lépéssel visszavonhatjuk. A tranzakciók támogatásának hiánya nem jelentett volna ugyan feltétlenül gondot az alkalmazás készítésekor, a jövőre azonban még ilyen kis alkalmazásnál is gondolni kell, így más rendszer után néztem, mely a tranzakciókat is kezeli.

Meg kell említeni, hogy a MySQL az idegen kulcsokat is csak korlátozottan támogatja. Az idegen kulcs a relációs adatmodell fontos

elemét képezi. Azt a módot jelenti, ahogy a viszonyokat ábrázoljuk, amolyan ragasztóként is felfoghatjuk, mely a táblákat együtt tartja, hogy azok egy relációs adatbázist formálhassanak.

Egy másik területén is gyengének találtam a MySQL-t, a részlekérdezéseket ugyanis nem támogatja. Ezzel a szolgáltatással a fejlesztő egy SQL kifejezést egy másikba ágyaz bele. Ismétlem, hogy alkalmazásom kicsi ugyan, és valószínűleg nem okoztak volna túl nagy gondot a MySQL említett hiányosságai, mégis úgy döntöttem, másik RDBM-et keresek.

PostgreSQL

A MySQL kapcsán felmerült hiányosságokat a jelek szerint pótolta a PostgreSQL (➔ <http://www.postgresql.org/>) fejlesztői csapata. A PostgreSQL 6.5.3 változata a RedHat 6.2-es kiadásában is megtalálható. Ez a változat már támogatja a tranzakciókat, a részlekérdezéseket, bár az idegen kulcsok terén csak korlátozottak a képességei.

A PostgreSQL leírása szerint a PostgreSQL a nagy, kereskedelmi DBMS-ekhez hasonlóan már a tranzakciók, a részlekérdezések, a triggerek, a nézetek és a kifinomult zárolások használatának lehetőségét is biztosítja. Van néhány olyan szolgáltatása is, amelyeket

más szabad adatbázis-rendszerek nem nyújtanak. Ilyenek például a felhasználó által meghatározott típusok, az öröklés, a szabályok és a többváltozatú versenyhelyzet-kezelés a zárolások miatt jelentkező torlódások csökkentésére. Ez a bővebb szolgáltatáskészlet sajnos nem biztosítható a MySQL-ével megegyező sebességgel. Összehasonlítva a MySQL-lel vagy a tanulásra képes adatbázis-rendszerrel, a PostgreSQL lassabb a frissítések és beillesztések végrehajtásánál, mivel a tranzakciók miatt jelentkező többletterhelést is kezelnie kell. Hasonlóan a MySQL-hez és éles ellentétben az Oracle-lel a PostgreSQL legkisebb rendszerkövetelményei szerények. A PostgreSQL rendszerfelügyeleti kézikönyve szerint, bár a PostgreSQL futtatásához akár 8 MB memória is elegendő lehet, észrevehető sebességnövekedés tapasztalható, ha 96 megabájt vagy annál is több memóriát használ. Az alapszabály az, hogy memóriából sosem lehet túl sok. Azt is ellenőrizni kell, hogy elegendő terület áll-e rendelkezésre a merevlemezen. A fordítás alatt körülbelül harminc megabájt szükséges a forrásokhoz, és nagyjából öt megabájt a telepítési könyvtárhoz. Egy üres adatbázis körülbelül egy megabájt helyet foglal, egyéb esetekben hozzávetőlegesen ötször akkora, mint amekkora a benne tárolt adatok mérete egyszerű, szöveges állományban lenne. Ha az időutazásos ellenőrzést is futtatjuk, átmenetileg további húsz megabájt helye is szükség lesz.

Az általam használt számítógép bőven megfelelt ezeknek a követelményeknek, így a sebesség csökkenésétől nem kellett tartanom. Elégedett voltam azzal, ahogy a PostgreSQL megoldja a MySQL- és Oracle-termékek kapcsán felmerült gondjaimat, tehát úgy döntöttem, hogy tervem megvalósításához a PostgreSQL-t használom.

Parancsnyelv választása

Az adatbázisokhoz hasonlóan a parancsfájlok készítésére alkalmas programnyelvekből is rengeteg elérhető a Linuxhoz. Ezek közül több is használható kiszolgálóoldali programok készítésére dinamikus honlapokhoz.

ColdFusion

A kereskedelmi termékek közül az egyik lehetséges választás az Allaire cég (☞ <http://www.allaire.com>) ColdFusion rendszere volt. Az Allaire termékét széles körben használják, a környezet kiszolgálóoldali része teljes támogatást élvez és egyaránt elérhető Linux és Windows NT-rendszereken. Az Allaire a ColdFusion Studiónak nevezett csomag részeként nagyon csinos fejlesztőkörnyezetet kínál, melynek futtatásához Windows 9x vagy NT operációs rendszer szükséges. Mindkét összetevő letölthető kipróbálásra, illetve megvásárolható az Allaire honlapján.

A ColdFusion csomag összetevőit a ColdFusion Markup Language (CFML) támogatására készítették. A CFML formai követelménye nagyon hasonló a HTML-éhez, nyitó és záró jeleket használ, melyek azonban a rendes HTML-énél sokkal bővebb lehetőségeket teremtenek. Ezek a különleges jelek keveredve vagy beágyazva jelennek meg a HTML alkalmazásban, illetve oldalon.

A ColdFusion kiszolgáló együttműködik a webkiszolgálóval, elfogja ezeket a különleges jeleket, így teszi lehetővé az alkalmazás háttérkövetelményeit biztosító adatbázissal vagy kiszolgálóval folytatott párbeszédet.

A termékről és a környezetről szerzett benyomásaim rendkívül kedvezőek. Hallatlanul jó felépítésű alaprendszernek tűnik, melyet könnyen lehet telepíteni és használni. A CFML és a HTML közötti hasonlóságok szintén segítik a kezdőket, hogy hamar munkára foghassák alkalmazásukat. Emellett a ColdFusion együttműködik az Apache kiszolgálóval, ami újabb jó pontot jelent. Vannak azonban hátrányai is, például az, hogy a ColdFusion kiszolgáló linuxos változatának rendszerkövetelménye 512 megabájt memóriát. Ez messze meghaladta az általam használt gép adottságait. A 2000. július 17-én az Allaire honlapjára feltett árlista alapján a ColdFusion Server 4.5 Professional for Linux termék ára 1295 dollár volt. A Windows-alapú fejlesztési környezet további 495 dollárba került. Ha alkalmazásunk költségvetése elviseli a ColdFusion megvásárlásának terhet, megfontolandó választás.

Perl

Azon túl, hogy a Perl internetes parancsfájlkészítő nyelvként használják, kevés olyan dolog van, amit a Perl nem tud kezelni. A nyílt forrású Perl *Larry Wall* készítette. Először 1987-ben jelent meg, az 5.6-os számot viselő változata pedig 2000. márciusában. Letölthető a ☞ <http://www.perl.org/> honlapról.

A Perl rendkívül jól átgondolt és életképes választás internetes programok készítésére. Fejlesztői közössége hatalmas, és remek támogatás érhető el hozzá. Képes szorosan együttműködni az Apache kiszolgálóval, és elérhető majdnem az összes Linux-változattal. Csakhogy én azt reméltem, hogy valamilyen egyszerűbb csomaggal is meg tudom oldani weblapjaim adatbázishoz kapcsolását. Az, hogy a Perl a legkézenfekvőbb megoldásnak tűnt, még nem jelentette azt, hogy egyben a legjobb is. Az egyszerűség követelménye jelentős szerepet játszott a programnyelvek vizsgálatokor, és ez eltért a Perl választásától. Alkalmazásomat minél hamarabb, a lehető legrövidebb tanulási időszak után el akartam készíteni.

PHP

A PHP (☞ <http://www.php.net/>) nyílt forrású, HTML-be ágyazott parancsnyelv. A Perllel ellentétben – amit a rendszerfelügyelet segítségére készítettek –, a PHP-t kezdetektől fogva honlapokkal végzett munkához tervezték. A PHP számos régebbi programozási nyelvből kölcsönöz elemeket, így a Perlből és a C-ből is. A PHP leírása a PHP és a Perl közötti különbségeket is tárgyalja. A PHP legnagyobb előnye a Perllel szemben éppen az, hogy a PHP-t internetes oldalak parancsfájljainak készítésére tervezték, míg a Perl számos egyéb feladatra is, ezért meglehetősen bonyolult lett. A Perl

rugalmassága és összetettsége folytán használatával könnyen készíthető olyan kódok, melyeket másik fejlesztő vagy olvasó csak nagyon nehezen ért meg. A PHP megőrzi a rugalmasságot, de formátuma kevésbé zavaró és kötött. Könnyebben együttműködik a meglévő HTML oldalakkal, mint a Perl. A PHP a Perl szinte összes kellemes tulajdonságát magában hordozza, a szerkezetek és a formai követelmények terén egyaránt, ám eközben nem válik olyan bonyolulttá, mint amilyen a Perl lehet.

Hasonlítsuk össze a PHP-t és a ColdFusion-t! A PHP leírása remek összefoglalást ad, és hivatkozik a két csomag *Mike Sheldon* által készített összehasonlítására is. Mike elemzése (☞ <http://marc.theaimsgroup.com/>) alapos munkának látszik. Kiemeli a ColdFusion néhány olyan erősségét és szolgáltatását, melyeket én nem tárgyaltam. A PHP két, számomra meglepő területen gyengélkedik: a hibák és a dátumok kezelésében. Az alkalmazás egyszerűsége miatt úgy gondoltam, ezek a hiányosságok számomra nem jelenthetnek különösebb hátrányt. Mindössze egyetlen adatbázist akartam használni, és az alkalmazás esetében minden adatbázishoz meghatározható egy rendeltetés-halmaz. A ColdFusion által nyújtott adatelvonatkoztatási réteg előnyeinek ez esetben nem láttam hasznát. A végső választásom négy összetevőt tartalmazott: GNU/Linux operációs rendszert, PostgreSQL relációs adatbázis-kezelő rendszert, PHP kiszolgálóoldali internetes parancsnyelvet és az Apache webkiszolgálót.

Összegzés

A Linuxhoz milliányi programmegoldás érhető el. Azt ajánlom, hogy akik hasonló programcsomag telepítését fontolgatják, alaposan gondolják át, vessék össze a programkínálat alkotóelemeinek erősségeit és fogyatékosait, melyek abból fakadnak, hogy valamilyen különleges igényt elégítenek ki. A fejlesztők és tanácsadók értéke csak részben származik abból, hogy egy-egy megoldást szinte tökéletesen ismernek. Azoknak a folyamatoknak a megértése, melyek várhatóan hatással vannak a megoldás elkészítésére, elsődleges fontosságú lehet. Mivel rengeteg a kiváló minőségű lehetőség egy-egy számítástechnikai kihívás megválaszolására, a szakértő feladata ezeknek a megoldásoknak a kezelése, ami kulcsfontosságú lehet a teljes tervezet megvalósításának sikere vagy sikertelensége szempontjából.

Saját tervemnél a Linux – Apache – PostgreSQL – PHP összeállítás tűnt megfelelőnek. Ugyanez az összeállítás más tervekhez nem feltétlenül megfelelő. Ennek ellenére remélem, hogy saját választásom szempontjait ismertetve sikerült segítséget nyújtanom néhány olvasó következő tervének megvalósításához és a különféle megoldások vizsgálatához.

Irodalomjegyzék

- Castro, E. (2000). *HTML for the World Wide Web: Visual Quickstart Guide*, fourth edition, Berkley, CA: Peachpit Press.
Kientzle, T. (2000). „Database Engines: MySQL versus Oracle”. *Dr. Dobb's Journal*, 25, 7, 98–104.
Loukides, M. and Oram, A. (1997). *Programming in GNU Software*. Cambridge: O'Reilly.
Watson, R. (1999). *Data Management: Databases and Organizations*, second edition, New York: John Wiley & Sons, Inc.



Dennis Roman Gesker

(drgst30@katz.pitt.edu) jelenleg a Pittsburghi Egyetemen tanul. Az oklevél megszerzése után a montanai Kalispell Alamon Telco cég vezetéséhez csatlakozik majd.