© Kiskapu Kft. Minden jog fenntartva

# A felhasználói viselkedés vizsgálata (2. rész)

A sorozat előző részében megismerkedtünk a digitális Nagy Testvér módszereinek elméleti lehetőségeivel. Tudjuk tehát, hogy mit és miért lehet és érdemes megvalósítani. Most lássuk hogy hogyan!

## Adatgyűjtés általában

Az internet, azon belül is a weblapok hálózata folyamatosan, és megdöbbentő mértékben növekszik, akár a forgalmat, akár a weboldalak összetettségét és számát vizsgáljuk. Ezzel a folytonos növekedéssel természetesen a weboldalakon látható grafikának, a webkiszolgálók tervezésének, kialakításának, méretezhetőségének is lépést kell tartania. Nem kivétel ez alól a weboldalakon belüli navigáció összetettsége sem, éppen ezért a webes fejlesztési folyamatok egyik legfontosabb „bemenő paramétere" a weboldal használatának jellemzése.

A weben elérhető információ mennyisége és szerkezeti összetettsége már régóta tökéletesen lehetetlenné teszi, hogy a hely pontos ismerete nélkül, egyszerűen csak megtaláljuk azt, amit éppen keresünk. Ezt felismerve kezdték el fejleszteni az adatbányászat (data mining) mintájára a *„web mining"* módszereket. A ***web mining*** szolgál az interneten fellelhető hasznos információ felfedezésére, elemzésére és kategorizálására. A web miningnak nevezett tevékenység tulajdonképpen maga is három fő részterületre osztható:

- ***Tartalom (Content Data)***: a felhasználónak jelentéssel bíró adatok elemzése. A ***„Web Content Mining"*** az a folyamat, amely során a webes dokumentumokból „kiemeljük" a tényleges tudást.
- ***Szerkezet (Structure Data)***: azoknak a metaadatoknak az összessége, amelyek meghatározzák egy szervezet webes információs rendszerének logikai struktúráját. A ***„Web Structure Mining"*** során a tudást (információt) tehát az adott honlap szerkezete hordozza.
- ***Használat (Usage Data)***: olyan adat, amely a felhasználók webes interakciói során keletkezik. A ***„Web Usage Mining"*** célja az, hogy szabályosságokat fedezzen fel a felhasználók viselkedésében, lépéseiben, vagy például abban, hogy mit töltenek le.

A szakirodalom a felhasználói viselkedés vizsgálatát „Web Usage Mining", míg a modellezéshez szükséges adatok előállítását „Usage Mining" néven említi.

Amint azt az előző részben már említette, a viselkedés modellezése alapvetően két részből áll: az adatgyűjtésből és az adatok elemzéséből. A gyűjtés feladata olyan adatok előállítása, amelyek megfelelően részletesek ahhoz, hogy az analízist végző szoftverek használható modelleket alkothassanak belőlük. Mivel nem célunk az, hogy minden egyes látogató viselkedését egyenként elemezzük, csupán az „átlagos látogató" paramétereire vagyunk kíváncsiak, ezért a megszemélyesítést lehetővé tevő adatokra (tipikusan IP cím és felhasználónév) csak a felhasználói munkamenetek elkülönítéséhez van szükségünk.

Amennyiben a felhasználókat más módszerrel is meg lehet különböztetni (például rendelkeznek egyedi munkamenet azonosítóval), akkor semmilyen személyes adatra nem lesz szükségünk, hiszen elegendő az oldal neve és megtekintésének időbélyege.

Az adatgyűjtési módszerek alapvetően három csoportra oszthatóak. A megfigyelést végző kódot beépíthetjük magába az oldalba, illetve végezhetünk adatgyűjtést az ügyfélen és a kiszolgálón. Mint megannyi más területen, itt is mindegyik módszernek megvannak a maga előnyei és hátrányai, amelyekkel tisztában kell lennünk, mielőtt alkalmazni kezdjük őket. Lássuk tehát!

## Weboldalba épített adatgyűjtés

A módszer lényege, hogy már a portál tervezésekor kiemelt figyelmet szentelünk annak, hogy a felhasználók tevékenységéről megfelelő információval rendelkezzünk. Gyakori megoldás, hogy a weboldal elérése regisztrációhoz kötött, és csak a regisztrált és beléptetett felhasználók érhetik el a hasznos információt hordozó belső oldalakat.

Ez a módszer komoly fejlesztési ráfordítást igényel, hiszen már a tervezést is számos ponton befolyásolja.

Ha nem a kellő körültekintéssel járunk el, az később hátrányosan befolyásolhatja a rendszer fejleszthetőségét, vagyis éppen azt, amiért az adatokat gyűjtjük, és a rendszer használatát modellezzük. Hiába tudjuk az adatok alapján, hogy milyen szerkezeti változásokat kellene végrehajtanunk, ha a gondatlan tervezés miatt nem tudjuk azokat elvégezni.

Az adatgyűjtést végző programkódok alig, vagy egyáltalán nem újrahasznosíthatóak, hiszen minden egyes weboldal felépítése – ha olykor igen csekély mértékben is de – eltérő. Ugyanakkor éppen azért, mert a kódot maga az oldal tartalmazza, egészen pontos információkat kapunk külön-külön minden egyes felhasználó viselkedéséről. Ez még akkor is

© Kiskapu Kft. Minden jog fenntartva

igaz, ha az illető a regisztráció alkalmával valótlan adatokat adott meg. Ráadásul megfelelő kialakítás esetén ez a módszer képes rögtön előszűrt adatokkal ellátni az analizálást később elvégző alrendszert.
A regisztráció és belépési kötelezettség azonban egyrészt elriaszthatja az anonimitásukat megőrizni igyekvő felhasználókat. A legtöbb ember a magánszférába való túlzott behatolásként értékeli az ilyen jellegű működést, illetve zavarja az a tudat, hogy egy lelketlen rendszer minden lépését figyeli. Megjegyzendő persze, hogy a portál megfelelő tartalommal ellensúlyozhatja a magánszférába való behatolást. A magánszféra megsértésének problémáját ki lehet küszöbölni úgy is, ha nem az egyes felhasználók műveleteinek sorozatát mentjük, hanem csak azt, hogy a felhasználóink mit csináltak a portálon belül. Ilyenkor az adatokat olyan formában tároljuk, hogy azokból eleve lehetetlen legyen megmondani mondjuk azt, hogy személy szerint Teszt Tamás merre barangolt az oldalak között. Természetesen ehhez az egészhez az is fontos, hogy a látogatók valamelyest megbízzanak a portált működtető szervezetben, hiszen a programkódokba nem lát bele az egyszerű felhasználó. Nyílt forrású szoftverrel ez a probléma részben feloldható, de az adatvédelmi elvek betartásának ellenőrzése komoly programozási ismereteket igényelhet.
Mint említettem, ennek a megoldásnak az is előnye lehet, hogy eleve szűrt adatokat szolgáltathat. Pontosan tudjuk naplózni, hogy mely weboldalt mikor kérték le, az eseménynapló közvetlenül feldolgozható és a munkamenetek elkülönítésével sem kell foglalkoznunk, mert a belépéskor minden egyes felhasználónak létrejön az egyedi munkamenet azonosítója. Mivel a weboldalt előállító program a kiszolgálón fut, a módszer rendelkezik a szerver oldali adatgyűjtés összes ismert hátrányával is.

## Adatgyűjtés az ügyfélen

Az ügyfél oldalán történő adatgyűjtés megpróbálja (bizonyos tekintetben szinte teljes sikerrel) a HTTP protokoll és a web felépítéséből, működéséből adódó hátrányokat kiküszöbölni.
Az alapvető módszer az, hogy az ügyfélen egy a böngészőtől független program (esetleg annak egy kiegészítése, vagyis egy „plug-in”) folyamatosan figyeli a felhasználó által végzett műveleteket: melyik weboldalt mikor kérte le, mennyi idő alatt érkezett meg az a kiszolgálóról, sőt akár az is mérhető, hogy az adott oldalt tartalmazó ablak mennyi ideig volt aktív.
Ez a bizonyos kiegészítő program többféleképpen kerülhet az ügyfélhez. Lehetőség van arra, hogy a honlap letöltésekor automatikusan betöltődjön és elinduljon a kliens böngészőjében úgynevezett beágyazott HTML objektumként. (Ilyen például egy JAVA applet, egy Flash, vagy egy ActiveX komponens.) Biztonsági okokból ez a megoldás csak korlátozottan használható. Az így letöltött program csak az adott (ahonnan letöltötték) kiszolgálóval tud kommunikálni, és nem képes a helyi géppel adatokat cserélni, hiszen csak ideiglenes fájlokat tud létrehozni, amelyek a portál elhagyásakor, de legkésőbb a böngésző bezárásakor megsemmisülnek. További probléma, hogy csak annak a portálnak a tartalmával kapcsolatban működőképes, ahonnan letöltötte a felhasználó. Utóbbi probléma részben kiküszöbölhető azzal, ha minden weboldalról letöltődik a program és egy közös helyen (például egy sütiben) tárolják a munkamenet azonosítót. További, nem elhanyagolható hátrány, hogy a sütikhez hasonlóan ez a lehetőség is letiltható a böngészőben.
Egy másik lehetőség az, ha egy a böngészőtől független, harmadik szoftver segítségével figyeljük a felhasználó webes műveleteit. Ekkor lehet a legpontosabb adatokat összegyűjteni, hiszen ez a harmadik, kiegészítő szoftver képes a böngészés előtt, alatt és után is rögzíteni a történéseket. Pontos adatokat kaphatunk a hálózati átviteli időkről, az egyes weboldalak olvasásának idejéről, megfigyelhető és mérhető a lokális és távoli átmeneti tárak (proxy kiszolgálók) hatása. Előny, hogy az így keletkező adatokat a későbbiek során nem kell előfeldolgozni sem.
Mindkét esetben szükséges, hogy a harmadik, kiegészítő program teljesítményigénye minimális legyen, hiszen ellenkező esetben a felhasználó hamarabb elhagyja a portált mert az lassú, illetve törli a programot a számítógépéről.
A módszer legnagyobb hátránya, hogy felhasználói aktivitást követel meg: magának a felhasználónak kell telepítenie és beállítania a szoftvert. További nehézség, hogy a felhasználók többségének nem fűződik érdeke az ilyen jellegű programok használatához, sőt egyre jellemzőbb a kifejezett elzárkózás.
Az ilyen programok általában nem szabványos protokollon és portokon keresztül juttatják vissza az összegyűjtött adatokat a kiértékelést végző rendszerhez, ezért könnyen előfordulhat, hogy mind az egyéni, mind a vállalati szférában egyre gyakoribb proxy és tűzfal megoldások nem engedik át az ilyen jellegű adatfolyamokat. Jellemző továbbá, hogy elsősorban vállalati környezetben a felhasználók jogosultságai igen korlátozottak, ezért nem tudnak semmilyen szoftver telepíteni.

## Adatgyűjtés a kiszolgálón

A kiszolgálón történő adatgyűjtés nem igényli az ügyfél együttműködését, így nincs szükség sem harmadik fél által írt programokra, sem a felhasználó hozzáértésére és hozzájárulására. A felhasználó megkímélhető a regisztrációtól is, mert az eseménynapló a portáltól független módon keletkezik.
A szerver oldali adatgyűjtés tulajdonképpen nem más, mint az általában amúgy is működő naplózó szolgáltatás (hibakeresés, teljesítmény analízis, behatolási kísérletek felderítése) kiterjesztése a tárolt speciális paraméterekkel (böngésző típusa, előző oldal (referer), munkamenet azonosító). Az így keletkező adatsor nem alkalmas arra, hogy valós időben szolgáljon adatokkal a modellezéshez, hiszen jelentős „tisztítást” igényel éppen a szerteágazó adattartalom miatt.
A módszer hátrányai a HTTP protokoll és a világháló működéséből adódnak.
A felhasználók minél gyorsabban szeretnék letölteni a tartalmat, a vállalatok viszont igyekeznek spórolni a kommunikációs költségekkel, illetve egyes tartalmakat szűrni akarnak. (Munkaidőben például ne magánjellegű információk keresésével töltse az idejét a munkavállaló.), Éppen ezért használnak annyian átmeneti tárakat és különböző proxy technikákat.
A modern böngészőprogramok azzal is segítik a felhasználót, hogy az egyszer már letöltött weboldalakat tárolják. Előfordulhat akár az is, hogy a felhasználó a weboldalba

épített navigációs elemeket használja (linkekre kattint) és nem a böngésző „vissza” és „előre” gombjait, az eseménynaplóban mégsem jelenik meg a bejegyzés, mert a böngésző szoftvere észleli, hogy letöltött honlapról van szó és a lokális átmeneti tárból szolgálja ki a kérést.
Az ilyen „zajok” kiszűrése nem lehetséges pusztán az eseménynapló feldolgozásával. A zaj jellege és mérete nem becsülhető, hiszen az ügyfélen végzett adatgyűjtéstől eltérően itt nem áll rendelkezésre pontos adat a tényleges lekérésekről. A probléma ugyanakkor részben áthidalható a webszerver beállításával. A HTTP protokoll lehetőséget ad arra, hogy az átküldött fejlécben jelezzük: nem akarjuk, hogy a kérdéses weboldal a lokális tárba kerüljön. Ilyenkor a böngésző ugyan minden alkalommal újra lekéri azt, ugyanakkor elvesznek a lokális és távoli átmeneti tárak, illetve proxy kiszolgálók által nyújtott előnyök.
A fentiek miatt a kérések egy meghatározhatatlan része el sem jut a webkiszolgálóig, jelentősen rontva ezzel a módszer pontosságát. Ugyanakkor az átmeneti tárakban leggyakrabban a képek tárolódnak, amelyek a viselkedés modellezés során nem (feltétlenül) jelentenek értékes információt. Mindent egybevetve a legjobb megoldás talán a három módszer valamiféle összeházasítása lehetne.
A kiszolgálón történő adatgyűjtés egy másik jellegzetes problémája a nem determinisztikus hálózati működés. Ez azt jelenti, hogy egy az eseménynaplóba bekerült felhasználói kérés jelenléte nem jelenti teljes biztonsággal azt, hogy a felhasználóhoz el is jutott a kért tartalom. Akár az is lehetséges, hogy a letöltés befejeződése előtt a felhasználó bezárta a böngészőt, hiszen a webkiszolgáló erről nem kap visszajelzést. Tekintettel a mai hálózatok viszonylagos megbízhatóságára az így keletkező zaj csak kis jelentőséggel bír.
Az eseménynaplóba minden a kiszolgálóhoz érkező kérés bekerül: minden lekért dokumentumnak, képnek, videónak, zenének, animációnak, appletnek, vagy fájlnak nyoma van. Ez azt jelenti, hogy egy-egy weboldal letöltésével akár több tíz bejegyzés is keletkezhet. Ugyanakkor a modellezés szempontjából lényeges információt csak bizonyos elemekkel kapcsolatos bejegyzések hordoznak. Ezért az eseménynaplót a portál szerkezetének (vagy a kívánt modellnek) megfelelő előfeldolgozás során „meg kell szűrni”.
A modern portálfejlesztés bevált módszere, hogy a különböző weboldalakat egyetlen fájl készíti el paraméterektől (esetleg a munkamenetben tárolt adatoktól) függően. Ekkor bár a felhasználó folyamatosan egy bizonyos weblapot kér le, a tartalom mindig más. A modellezés során ugyanakkor nyilván el kell különíteni ezeket a lekéréseket, hiszen tartalmilag különböző weboldalaknak tekintendők. Az előfeldolgozás során az eseménynapló lekérések mezőjének vizsgálatával lehet elkülöníteni a formailag azonos fájlhoz tartozó, de tartalmilag különböző lekéréseket.
A POST metódussal átadott és a munkamenetben tárolt paraméterek esetén újabb problémával szembesülünk. Az ilyen lekérések sajnos nem kerülnek be az eseménynaplóba és általánosan használható módszerek sincsenek a mentésükre. Nem is szokás tárolni ezeket a kéréseket, hiszen egy-egy ilyen kérés mérete akár több megabájt is lehet, ami egy komoly forgalmú portálnál a tartalomnál akár négy-nyolc nagyságrenddel nagyobb méretű eseménynaplót eredményezne.

Mindez ugyanakkor igazi problémát csak akkor jelent, ha a POST illetve munkamenet adatok ugyanarra az oldalra kerülnek vissza rendszeresen, mint ahonnan elindították a kérést. Egyre több szakember van azon a véleményen, hogy egy URI-nak könnyen olvashatónak és érthetőnek kell lennie, vagyis legalább körülbelül ki kell derülnie belőle, hogy mi található az adott weboldalon.
Régóta biztosítják a webszerverek a kérések átírásának lehetőségét, amellyel így részben elfedhetők a POST és GET metódus miatti azonos lekérések.

Ha pédául a

```
http://www.ceg.hu/index.php?m=0&sub=1&c_id=2&r_id=3
➥&act=5
```

forma helyett a

```
http://www.ceg.hu/szolgaltatasok/hardverfejlesztes/
➥ajanlat
```

formát használjuk, akkor az átalakítást a webszerver végzi a megfelelő beállítások szerint. Az ilyen úgynevezett ***„barátságos linkek”*** nagyban megkönnyítik a felhasználói viselkedés modellezését, mert az eseménynaplóban egyértelműen és könnyen szétválogatható módon jelennek meg a vizsgálandó elemek.
Az előfeldolgozás sajnos jelentős többletterhelést jelent a modellkészítés során, így ez a módszer nem alkalmas arra, hogy online, azaz folyamatosan frissülő statisztikákat és modelleket szolgáltasson. Tipikus megoldás, hogy a feldolgozás és modellalkotás a portál legkevésbé terhelt időszakában napi rendszerességgel zajlik (általában éjszaka).
A modellezés során a legnehezebb feladat az egyes felhasználók elkülönítése. Az eseménynapló alapesetben nem tartalmaz ehhez elégséges információt, mert a proxy és tűzfal mögül érkező látogatók adatai szinte teljesen megegyeznek. Egyes webszerverekhez már létezik olyan kiegészítő modul, amely munkamenet azonosítót rendel minden felhasználóhoz. Használható ezen kívül idő alapú elkülönítés, user agent vizsgálat, illetve a referer és lekérések közötti folyamatosság vizsgálata a portál szerkezetének ismeretében.
A módszer előnye, hogy sem hardveres sem szoftveres együttműködést nem igényel az ügyféltől. Csupán a sütik támogatása szükséges a munkamenet azonosító tárolásához, nem kell azonban sem az aktív felhasználói közreműködés, sem harmadik program vagy regisztráció.
Adatvédelmi szempontból is kedvezőbb megoldás a már ismertetetteknél, ugyanis a felhasználó nem kerül olyan mértékben megszemélyesítésre, ami az a magánéletét túlzottan sértené. A ma jellemző tűzfal, proxy és dinamikus IP cím kiosztási technikák tovább gyengítik a megszemélyesíthetőséget. Az eljárás egyedüli, ám igen lényeges hátránya, hogy a felhasználó tudta és beleegyezése nélkül is keletkeznek a tevékenységével kapcsolatos adatok.
Következő cikkemben részletesen ismertetem a szükséges Apache modulokat, illetve azok beállításait.

***Beszédes Balázs***

© Kiskapu Kft. Minden jog fenntartva