A felhasználói viselkedés vizsgálata (4. rész)

A sorozat előző részében megvizsgáltuk, hogyan juthatunk a modellezéshez szükséges adatokhoz. Most egy egyszerű program segítségével bemutatom az alkalmazott algoritmusokat és azok különböző paraméterezési lehetőségeit.

Az eddigiekben körüljártuk az elméleti lehetőségeket: modellezési kérdések és modelltípusok, adatgyűjtési módszerek, felhasználók azonosítása és elkülönítése. Most nézzük meg, hogy hogyan lehet használható modelleket, adatokat készíteni. Ehhez egy ***PHP*** nyelven írt szkriptet használunk, illetve ezen keresztül vizsgáljuk meg a lehetőségeket és az eredményeket.

Követelmények

Néhány elérhető modellező szoftver vizsgálata után fogalmaztam meg a legfontosabb követelményt: minél többféle modell készítését támogassa az eszköz, az előállított adatsorokat lehessen később más szoftverrel is feldolgozni (grafikus megjelenítés, elemzés), azaz legyen felkészítve sokféle (elsősorban szöveges) formátumú kimenet előállítására.

Előszűrés

A megvalósítandó eszköz egyik legfontosabb része az előfeldolgozó. Az általános felhasználhatósághoz ennek az alrendszernek széles paraméterezhetőséggel kell rendelkeznie, hogy bármilyen portál rendszer eseménynaplóját fel tudja dolgozni.

Három elkülöníthető funkcióra van szükség:

- ***Kihagyás***. A modellezés során a portáltól függően rengeteg eseménynapló bejegyzés nem hordoz releváns (a modellezés szempontjából felhasználható) információt. Ezeket a bejegyzéseket ki kell venni, mert csak zavarnák és lassítanák a feldolgozás és modellalkotás menetét.
- ***Összevonás***. A webszerverek és a HTTP protokoll lehetőséget nyújt ún. átirányításokra. Ez azt jelenti, hogy ha a felhasználó adatokat küld a szervernek, az azt feldolgozó program a lefutása után gyakorlatilag észrevétlenül egy másik weboldalra küldi tovább a kliens böngészőjét. Másik lehetőség, hogy egy weboldal több keretből (frame) áll, ekkor az egyes keretekbe betöltődő weblapok mind külön-külön bejegyzéseket jelentenek az eseménynaplóban. Ilyen esetekben tehát szükség lehet több eseménynapló bejegyzés összevonására, és egy weboldalként való szerepeltetésére a modellezés során.
- ***Átalakítás***. Egy másik jellemző megfontolás, hogy a portál összes weboldala egy fájlon keresztül érhető el. Ekkor az eseménynaplóban is csak ez az egy fájl jelenik meg, de a HTTP kérésből kiolvashatóak a paraméterek. Ebben az esetben arra van tehát szükség, hogy az ilyen paraméterezett lekérésekből egyszerű, könnyen olvasható és a lényeget kiemelő „weboldalakat” (tartalmat jellemző, címet és nevet tükröző modell bejegyzéseket) csináljunk.

Az előszűrés feladata továbbá, hogy a feldolgozhatatlan eseménynapló bejegyzéseket is eltávolítsa. Ilyen bejegyzések például a hibák, a nem létező erőforrásokra mutató lekérések.

Felhasználói azonosítás

A megvizsgált szoftverek között van, amelyik képes az egyes eseménynapló bejegyzéseket elkülöníteni, és egy-egy látogatóhoz rendelni. Ezen szoftverek legnagyobb hátránya az volt, hogy sehol nem volt elérhető a használt módszer(ek) leírása. (Az általam használt módszerekről cikksorozatom előző részében volt szó). Ezen okból kifolyólag a legfontosabb követelmény az, hogy az eszköz képes legyen a lehető legtöbb módon ezt a hozzárendelést elvégezni, és az egyes módszerek paraméterei széles skálán legyenek állíthatók (ne legyenek „bedrótozva” a szoftverbe).
A felhasználók elkülönítésére használatos módszereket alapvetően meghatározza a rendelkezésre álló eseménynapló, ezért szükséges, hogy a szoftver ilyen képességét (felhasznált metódusokat) mindig az adott adatforráshoz lehessen igazítani.

A lehetséges módszerek:

- munkamenet azonosító alapján, amelyet a webszerver helyez el az eseménynaplóban;
- idő (két lekérés között eltelt idő figyelése) és kliens IP cím alapú hozzárendelés legegyszerűbb (legkevesebb adatot tartalmazó) eseménynaplók esetén;
- részletesebb eseménynaplók esetén a ***referer*** és ***user agent*** értékek is bevonásra kerülnek az összerendelés folyamatának pontosításába.

Az eseménynapló két bejegyzése akkor tartozik egy felhasználóhoz, ha a weboldal lekérések között eltelt idő értéke (idő paraméter) kisebb, és akkor különítendő el más-más látogatóhoz, ha nagyobb, mint a beállított idő paraméter értéke. Ezen idő paraméter értéke nagymértékben befolyásolja a létrejövő modelleket (látogatók száma, különböző útvonalak megoszlása, stb.), ezért kiemelten kell kezelni ennek beállítási lehetőségét.

Exportálás

A szoftverrel szemben nem követelmény, hogy szép formában (grafikusan, szemléletesen) állítsa elő az eredményt, de fontos cél, hogy az adatokat többféle szöveg alapú forrásba képes legyen menteni a további feldolgozhatóság miatt. A lehetséges formátumok például: ***text***, ***CSV (comma separated value)***, ***XML***, ***HTML***. Ez azért szükséges, hogy további modellező és megjelenítő eszközök felé biztosított legyen az átjárás, megkönnyítve ezzel az érthetőséget és az ellenőrzést.

Az analizáló szoftver

Az analizáló szoftver egymás utáni feldolgozási lépések folyamatából áll, ugyanakkor a további felhasználás és fejlesztés szempontjából érdemes egy osztályba összefogni az egyes lépéseket megvalósító kódokat. Ennek további előnye, hogy az OOP-szemlélet következtében az eseménynapló vizsgálatának eredménye egyszerűen beilleszthető egy másik szoftverbe is (például egy általános eseménynapló statisztikai programba).

A fentieknek megfelelően a szoftver két fájlból áll:

- A ***WebUserModeller.class.php*** tartalmazza azt az osztályt, amely egybefogja a modellezés lépéseit megvalósító függvényeket, a függvények által közösen használt változókat, elrejti a belső funkciókat (például: eseménynapló sor beolvasása) és megfelelő metódusokat (interfészeket) biztosít az igénybevételhez. A modellezést végző kódok és algoritmusok ebben az osztályban találhatóak meg.
- A ***wum.php*** használja az előzőekben ismertetett osztályt a vizsgálathoz. Egyes paramétereket a ***wum.php*** programkódjában lehet változtatni, ezeket a beállításokat ebben a fájlban lehet megadni (például: vágási küszöbértékek). Ez a fájl biztosítja jelenleg a modellező osztály használatának lehetőségét.

Az elemzőprogram a ***wum.php*** parancssori meghívásával indítható el a megfelelő paraméterek megadásával. Feladata a megadott paraméterek szerint (használt konfigurációs fájl, elkészítendő modell típusok, kimeneti könyvtár) a modellező osztály megfelelő metódusaink meghívása, és a létrejött eredmények feldolgozása, ami jelen esetben a fájlba mentést jelenti.

Konfigurációs lehetőségek

Minden eseménynaplóhoz készíteni kell egy konfigurációs fájlt, amely az adott naplófájlra jellemzően és a felhasználás módjától függően tartalmazza a modellezési paramétereket. Ezek a paraméterek alapvetően befolyásolják az elkészülő modelleket.

A következő paramétereket kell beállítani a konfigurációs fájlban:

- ***log_file***: az analizálandó eseménynapló elérési útvonala, kötelezően megadandó.
- ***base_href***: a portál alap linkje (URI).
- ***use_cookie***: a felhasználók azonosításához lehetőség van munkamenet azonosító használatára. Ellenkező esetben idő alapú elkülönítéssel választja el a szoftver a látogatókat. Alapértelmezésben nem munkamenet alapú azonosítás történik.
- ***root_page***: a weben általában egy portál nyitóoldalának eléréséhez elegendő egy tartomány címet (például: ***http://www.linuxvilag.hu***) ismerni, nem kell a pontos kezdő fájl teljes nevét is manuális megadni (például: ***http://www.linuxvilag.hu/index.html***), ugyanakkor a háttérben mindig van egy fájl. A két elérési lehetőség egybeolvasztásához lehet itt megadni a kezdő fájl nevét, és a szoftver automatikusan lecseréli a nyitóoldali lekéréseket az itt megadottra, így biztosítva a modellek egységességét. Elhagyása esetén nem kerül figyelembevételre.
- ***allow_without_extension***: hasonló a ***root_page***-hez, beállításával a modellezés során a könyvtár lekéréseket lehet kihagyni illetve bevonni az analizálásba. Alapértelmezésben megengedő állapotú.
- ***remove_get_params_from_request***: a kérésekben szereplő ***GET*** paramétereket lehet eltávolítani, így például a felhasználó adatküldéseit lehet kiszűrni, vagy a ***GET*** paraméterbe kódolt különböző oldalakat lehet egynek venni, ezáltal akár radikálisan csökkentve a modellek felbontását. Alaphelyzetben nem engedélyezett.
- ***time_window***: a legfontosabb paraméter, ezt az időt veszi alapul a szoftver a felhasználók elkülönítésénél. Ha a ***use_cookie*** be van kapcsolva, akkor ezen érték nem kerül felhasználásra. Megadása kötelező.
- ***request_file_types***: az analízis folyamat az itt megadott típusú fájlokat (lekéréseket) veszi csak figyelembe a modellalkotás során. Ha nem szükséges a képek lekéréseinek modellezése, akkor itt nem kell megadni az adott portálra jellemző képtípusokat. Azokat a fájltípusra jellemző karakterláncokat kell megadni felsorolva, amelyek szerepelhetnek az engedélyezett kérésekben. A szoftver egyszerű mintakeresést végez. Mivel ez a modellezés egyik alapbeállítása, ezért mindenképpen meg kell adni a feldolgozandó kérések jellegét.
- ***request_simplify***: a modellezést nem befolyásolja közvetlenül, azonban megkönnyíti a modell elemzését azáltal, hogy a megadott paramétereknek megfelelően az egyes lekéréseket egyszerűbb szövegre cseréli le (például: ***index.php?modul=bolt&kategoria=muszaki_cikk*** helyett csak ennyi lesz a kimeneti eredményekben: ***bolt-muszaki_cikk***). A cserélendő kérésre jellemző (vagy azzal megegyező) mintához kell megadni az új (cserélt) karaktersorozatot. Opcionális paraméter.
- ***groupize_path***: az előszűrésnél említett összevonásokat lehet itt megadni Az új, összevont kéréshez fel kell sorolni az összevonandó kérésekre jellemző mintákat. Opcionális.
- ***page_referers***: az analízis során a felhasználók elkülönítéséhez felhasználható a ***combined*** szerkezetű eseménynaplókban szereplő ***referer*** érték is. Itt lehet megadni,

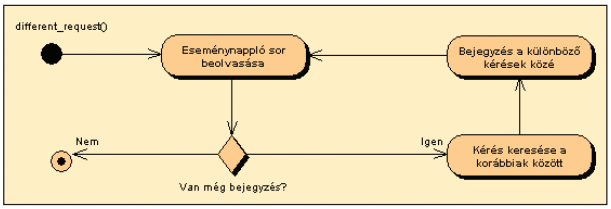
hogy az egyes weboldalakhoz mely másik weboldalak tartozhatnak ***refererként***. A portál weboldalaihoz felsorolásszerűen kell megadni a lehetséges ***referer*** értékek mintáit. Ha egy adott weboldalhoz nincs megadva lehetséges ***referer*** érték, az azt jelenti, hogy bármilyen ***referer*** értéket elfogadunk. Opcionális, csak a megadott weboldalaknál használja, vagyis minden egyes weboldalra külön-külön lehet megadni.

Használat, indítási paraméterek

Az analizáló szoftver képességei a ***wum.php*** indításával használhatók ki. Kötelezően megadandó paraméter a `--ini`, amely értéke határozza meg a felhasználandó konfigurációs fájl útvonalát.

A következő paraméterek opcionálisak:

- `--help`: minimális leírást ad a megadandó paraméterekről.
- `--output`: a kimeneti fájlok mentési könyvtára, ha nincs megadva, akkor a wum.php futtatási könyvtárába kerülnek az eredményeket tartalmazó fájlok.
- `--version`: a különböző beállításokkal készült modellezések elkülönítéséhez lehetőség van egy verziószám megadására, ez a verzió azonosító az eredmény fájlok nevét befolyásolja (hozzáfűzésre kerül), elhagyása esetén egy időbélyeggel helyettesítődik. Létező fájlnév esetén automatikus felülírás történik.
- `--diff`: az a küszöbérték, amely alatti előfordulásokat az egyes eredményekből el kell távolítani. A viszonyítás alapja a felhasználói útvonalak száma (vagyis az összes látogató száma). Megadása opcionális, alapértéke a 0, ekkor nem kerül figyelembe vételre.
- `--model`: a különböző lehetséges modellek elkészítését lehet bekapcsolni segítségével. Elhagyása esetén minden lehetséges modellezési folyamat lefuttatásra kerül.
- A `--model` paraméter lehetséges értékei és a hozzá tartozó modellek:
- ***up (user_path)***: A modellezés alapja, a felhasználók elkülönítését és az útvonalak összegyűjtését végzi. A további modelltípusok elkészítéséhez ennek a modellnek már rendelkezésre kell állnia.
- ***ro (routes)***: a felhasználók útvonalait állítja össze, megszámolja, hogy ugyanazokat a weboldalakat hány különböző látogató járta végig. Itt a weboldalak meglátogatási sorrendje is számít.
- ***sp (same_pages)***: abban különbözik a routes adatoktól, hogy itt már nem számít az oldalak sorrendje, tehát csak azt vizsgáljuk, hogy a felhasználók közül egy látogatás során hányan jártak azonos oldalakon (a teljes látogatást tekintve).
- ***ct (click_times)***: az egyes weboldalak közötti átkattintási időminimumokat, időmaximumokat és időátlagokat mutatja. További felhasználása: az itt kapott értékek alapján munkamenet azonosító nélküli, idő alapú eseménynapló elemzés.
- ***wmi (web_model_in)***: azt mutatja meg, hogy az egyes oldalakra honnan és hányszor érkeztek a felhasználók. A bejövő (tulajdonképpen az előző) oldalakra nincs szűrés, így nincsenek elkülönítve a kívülről illetve a portálon belülről érkezettek.



1. ábra Munkamenet azonosító alapú analízis

- ***wmo (web_model_out)***: hasonló a ***wmi*** modellhez, a különbség az irányban van. Ebben az esetben azt vizsgáljuk, hogy egy adott weboldalról melyik másik weboldalra léptek tovább a felhasználók.
- ***dr (different_requests)***: az eseménynaplóban megtalálható összes különböző kérést gyűjti ki. Nem igazi modell, segédeszköz a modellezés paramétereinek pontosításához. Önmagában is használható.

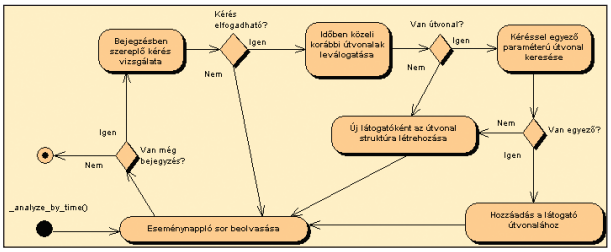
Működés, algoritmusok

Az analizáló szoftver lineáris lefutású: eseménynapló sorának beolvasása, feldolgozás, az adatok szerint egy felhasználó keresése vagy új létrehozása, majd a különböző modellek összeállítása a felhasználók adatsorai alapján.
Az analizáló szoftver legfontosabb része a felhasználók elkülönítéséért felelős rész, mert minden további modellezést végző részfeladat ennek az eredményein operál.

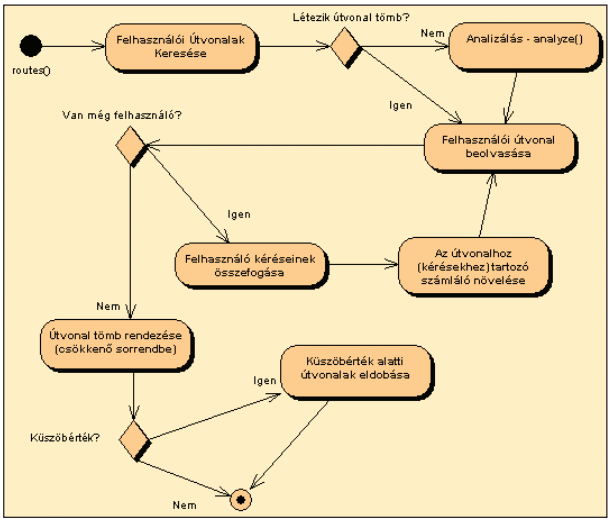
Felhasználók válogatása

- ***Felhasználók válogatása munkamenet azonosítók alapján***: A munkamenet azonosító alapú eseménynapló analízis folyamata egyszerű, mert minden egyes bejegyzésben már szerepel a látogatókat egyértelműen elkülönítő (azonosító) adat. Ez a módszer egyszerűen az azonosító alapján tömbbe gyűjti a látogatókat (*1. ábra*), és a hozzájuk tartozó weboldal lekérések paramétereit. A szoftver ebben az esetben nem veszi figyelembe az IP címet, időbélyeget, ***referer***t, stb., hiszen ezeket az adatokat a webszerver a weboldal lekérésekor „belekódolta" a munkamenet azonosítóba.
- ***Felhasználók válogatása idő alapú elkülönítéssel***: A látogatók elkülönítése és útvonalaik összegyűjtése elsődlegesen az IP cím, távoli eseménynapló, távoli felhasználó adatok alapján történik. Ez önmagában nem elegendő az egyértelmű megkülönböztetéshez. További segítséget nyújt a ***referer*** és a böngésző adatok jelenléte. Azonban a legegyszerűbb szerkezetű eseménynaplók nem tartalmaznak az IP címen és a kérés időpontján kívül több olyan adatot, amely segítené a látogatók elkülönítését, vagy egy nagyvállalati hálózatból érkező látogatók esetén jellemzően a böngésző adatok is azonosak.

Az idő alapú elkülönítés során (*2. ábra*) az éppen vizsgált kérést a megadott időintervallumon belül eső, kéréssel rendelkező látogatóhoz próbáljuk meg csatolni, oly módon, hogy a fent említett adatok közül a lehető legtöbb egyezzen. Teljes egyezés esetén a vizsgált kérést az adott látogatóhoz kapcsoljuk, ha nem találtunk teljes egyezést, akkor mint új látogató kezeljük, és egy új útvonal tömb bejegyzést nyitunk neki.



2. ábra Idő alapú analízis



3. ábra Különböző útvonalak modellezése

Felhasználói útvonalak

Az analízis eredményeképpen előálló felhasználónkénti lekéréseket (útvonalakat) mutatja meg. Az eredmény egy többdimenziós tömb, amely felhasználónként tartalmazza az egyes (sorrendben) lekért weboldalakat és néhány, az egyes kérésekhez tartozó kiegészítő adatot. Ez egy interfész, a tömb valójában az analizáló alrendszer eredménye.

Például:

```
[15] => Array
  (
    [0] => Array
      (
        [ip_address] => 213.157.96.206
        [remote_log] => -
        [user_id] => -
        [time] => 1050899398
        [request] => /index.php
        [referer] => http://www.jegyzet.com/
        [user-agent] => Mozilla/4.0 (compatible;
        ➥MSIE 6.0; Win98)
      )

[1] => Array
      (
        [ip_address] => 213.157.96.206
        [remote_log] => -
        [user_id] => -
        [time] => 1050899431
        [request] => /kereses.php
        [referer] => http://www.jegyzet.com/
        ➥index.php
        [user-agent] => Mozilla/4.0 (compatible;
        ➥MSIE 6.0; Win98)
      )
  )
```

A példa a „15-ös” látogató útvonalát mutatja: először lekérte az ***index.php*** weboldalt, majd a ***kereses.php*** weboldalt. Több kérése nem volt ennek a látogatónak. A példából kiolvasható, hogy melyik honlapot milyen időpillanatban kérte le, mi volt a kéréshez tartozó ***referer***, böngésző, kliens IP cím, stb. A következő folyamatok az ilyen tömbökből álló felhasználói útvonalak tömb alapján készítik el a specializált modellt.

Különböző útvonalak

A felhasználói útvonalakban (*3. ábra*) keresi meg az azonosakat (a bejárt weboldalak és sorrendjük azonos), ezen útvonalakat összegzi. Lehetőség van arra, hogy egy paraméterként megadott százalék alatti előfordulásokat eldobja, a viszonyítási alap a látogatók száma. Paraméterként megadható, hogy a megadott előfordulási százalék alatti útvonalak ne szerepeljenek az eredményben.
A módszer a következő: a felhasználói útvonalak tömbjét végigolvasva minden felhasználó útvonala leképzésre kerül egy egyedi karakterláncba. A megegyező útvonalakat (sorrend és többes látogatások is számítanak) bejárt felhasználóknál ez az egyedi karaktersorozat azonos. A továbbiakban a létrejött egyedi útvonalleírók kerülnek összegzésre (megszámoljuk, hogy egy egyedi útvonalat leíró karaktersorozatból hány van).

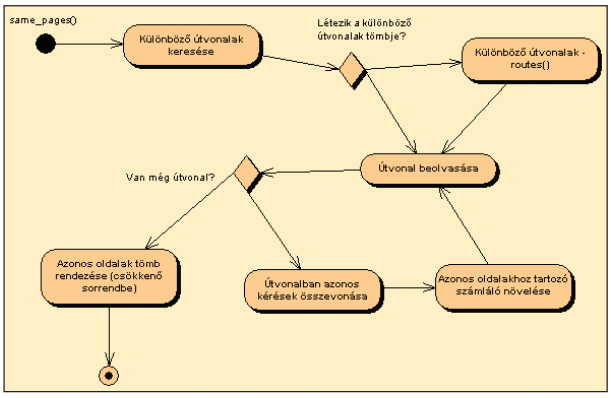
Például:

```
[/index.php] => 3128
[/index.php /eredmeny.php] => 1016
[/index.php /eredmeny.php /eredmeny.php] => 386
[/index.php /eredmeny.php /kereses.php] => 271
```

A fenti példában látható, hogy csak az ***index.php*** fájlt összesen 3128 látogató nyitotta meg, ugyanakkor az ***eredmeny.php*** weboldalt már csak 1016-an nézték meg. A weboldal lekérések sorozata különálló, vagyis a példa első sora nem tartalmazza a második sorból az ***index.php*** weboldalra irányuló lekéréseket.

Azonos oldalak egy látogatáson belül

Hasonló a különböző útvonalak módszerhez, azzal a különbséggel, hogy az egyes weboldalak meglátogatási sorrendje nem számít. Itt is lehetőség van a nagyon ritka előfordulások elhagyására.
A módszer (*4. ábra*) majdnem azonos a különböző útvonalak elkészítésének menetével. A különbség annyi, hogy itt a felhasználói útvonalakból az azonos weboldalra irányuló többes lekérések összevonásra kerülnek (egyszer szerepelnek), illetve nem számít az egyes lekérések sorrendje sem (ABC sorrendbe van rendezve a lekéréseket leíró karakterfűzér).



4. ábra Azonos oldalak egy látogatáson belül

Például:

```
[/index.php] => 3350
[/eredmeny.php /index.php] => 1947
[/eredmeny.php /index.php /kereses.php] => 1392
[/index.php /kereses.php] => 195
```

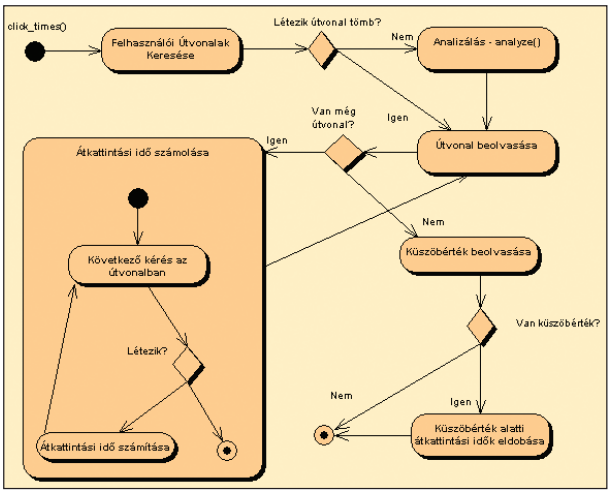
Átkattintási idők

Az egyes weboldalak közötti átkattintások számát, minimális, maximális, összes és átlagos idejét adja meg egy többdimenziós tömbben. Az átkattintásoknak irányuk van (forrás és cél), az eredményben külön van választva a két irány. Elsősorban az idő alapú látogató-elkülönítésnél fontos, az átlagos és maximális idők alapján lehet a megfelelő idő paramétert módosítani. Paraméterként megadható, hogy azok az átkattintások ne szerepeljenek az eredményben, amelyek száma nem éri el a felhasználók számának megadott százalékát.
A különböző átkattintási idők számítása (*5. ábra*) a felhasználói útvonalak vizsgálatával történik, az útvonal két lekérése közötti idők kerülnek vizsgálatra és összegzésre, átlagszámításra.

Például:

```
Array
(
  [/index.php] => Array
    (
      [/eredmeny.php] => Array
        (
           [clicks] => 3463
           [duration] => 121315
           [avg_dur] => 35.03 sec
         )
    )
)
```

A példa azt mutatja, hogy az ***index.php*** weboldalról összesen 3463 átkattintás történt az ***eredmeny.php*** weboldalra, a két weboldal lekérése között összesen 121315 másodperc telt el, amely adatokból már egyszerűen kiszámolható, hogy az ***index.php*** lekérése után átlagosan 35.03 másodperccel kérték le a felhasználók az ***eredmeny.php*** weboldalt.



5. ábra Átkattintási idők számítása

A modellezés inkább az általános érvényű megállapításokat keresi, ezért itt is lehetőség van a ritka átkattintások elhagyására, ezáltal egyszerűsítve a modell érthetőségét és áttekinthetőségét.

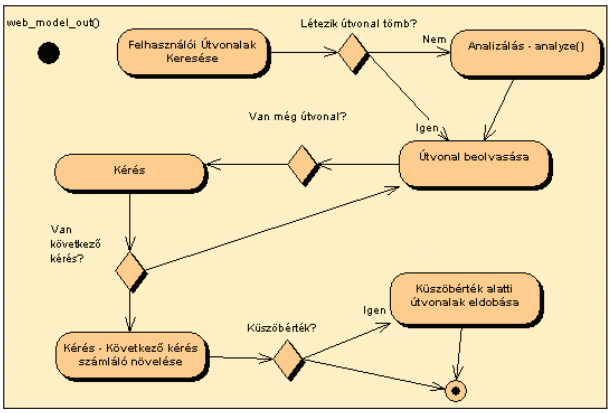
Be- és kilépő oldalak

A belépő weboldalak egy portálon belüli weboldalhoz adják meg, hogy honnan és hányszor nyitották meg (ezek között szerepel portálon belüli és kívüli forrás is). Ez az adatsor az analízis során létrejön, mert egyszerűen előállítható az eseménynapló bejegyzésből a ***referer*** adatok használatával. A belépő oldalak statisztikája csak akkor állítható elő, ha a ***referer*** adatok léteznek az eseménynaplóban, ellentétben a kimenő oldalak statisztikájával, ami a felhasználók lekérései alapján kerül összeállításra (útvonalban következő lekérés alapján), vagyis a legegyszerűbb szerkezetű eseménynapló alapján is előállítható.

Például:

```
[/index.php] => Array
  (
    [-] => 4979
    [http://www.oktaton.hu/index.php] => 1355
    [http://www.puska.hu/portal1.htm] => 373
    [http://www.lapkereso.hu/keres.php] => 350
  )
```

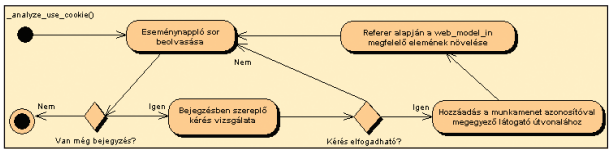
A fenti példában látható, hogy az ***index.php*** weboldalra 4979 felhasználó érkezett ***referer*** adat nélkül (ezt jelenti a „-”). Ez jellemzően abban az esetben fordul elő, amikor a felhasználó egy új böngésző ablakot nyit és annak címsorába maga gépeli be az adott portál elérhetőségét.
A kilépő weboldalak ellenkezőleg működnek, mert itt azt vizsgáljuk, hogy egy adott weboldalról merre mentek tovább a látogatók, ahol minden weboldal szigorúan az adott portálhoz tartozik. Ez a modell a felhasználói útvonalak elemzésével gyűjti össze az adatokat, megvizsgálja, hogy egy lekérés után melyik másik weboldal lett lekérve és összeszámolja az egyes párok előfordulásait (*6. ábra*).



6. ábra Kilépő oldalak statisztikák készítése

Például:

```
Array
(
  [/index.php] => Array
    (
        [/eredmeny.php] => 3774
        [-] => 3751
        [/kereses.php] => 640
        [/index.php] => 610
        [/reszlet.php] => 97
        [/figyeles.php] => 62
```



7. ábra Különböző lekérések kigyűjtése

```
        [/feladas.php] => 54
        [/tajekoztato/?t=ismerteto] => 15
        [/tajekoztato/] => 47
        [/tajekoztato/?t=szabalyzat] => 25
    )
)
```

Fenti példában a „-” jelentése: nem volt a felhasználónak további weboldal lekérése az ***index.php*** után, azaz távozott az adott portálról és az utolsó meglátogatott weboldala az ***index.php*** volt.
Mindkét esetben megadható az a százalékos paraméter, ami alatti előfordulásokat ki kell hagyni az eredmény tömbből.

Különböző kérések

Eltérően az eddig ismertetett analizáló és modellező alrendszerektől, ez a metódus inkább az előzetes eseménynapló vizsgálatot segíti azáltal, hogy a megtalálható összes különböző kérést kigyűjti (*7. ábra*). Így megvizsgálható, hogy a tényleges modellezéshez mely lekérések (weboldalak)

használandóak. Más szemszögből megtudható, hogy egyáltalán mely weboldalakat kérték le a látogatók a szerverről. A függvény használja az eseménynapló bejegyzéseket szűrő és egyszerűsítő folyamatokat, amelyek a konfigurációs fájlban akár ki is kapcsolhatóak.
Paraméterként megadható, hogy a kérések egyszerűsítés után kerüljenek összehasonlításra, vagy minden kérést úgy vegyen a szoftver, ahogy a naplófájlban szerepel.

Továbbfejlesztési lehetőségek

A megvalósított analizáló szoftver elsősorban az algoritmusok és lehetséges modellek bemutatására készült, az alapfunkciókat biztosan képes nyújtani. A fejlesztés során felmerült továbbfejlesztési lehetőségek ismertetése következik.

Gyorsítás, nagyméretű eseménynaplók feldolgozása

A első fontos fejlesztési lépés a szoftver működésének gyorsítása, amely összefügg a nagyméretű (200 MB feletti) eseménynaplók feldolgozásával. A szoftver jelenleg a teljes adatstruktúrát (felhasználói útvonalak) memóriában tárolja, ami legalább annyi memóriát igényel, mint a feldolgozandó eseménynapló mérete.
Tekintve, hogy már egy közepes forgalmú portálon is akár napi néhány száz megabyte eseménynapló keletkezik, megoldandó, hogy ne kelljen az egész struktúrát a memóriában tárolni. Elég volna csak azokat a felhasználói útvonalakat megtartani a memóriában, amelyek az éppen elemzett eseménynapló bejegyzés feldolgozásakor még számításba jöhetnek, mint olyan felhasználói útvonal, amelyhez az adott kérés hozzácsatolható (beilleszthető a látogató útvonalába). A figyelembevétel eldöntését megkönnyíti, hogy mind a munkamenet azonosító alkalmazása, mind az idő alapú látogató elkülönítés esetén behatárolt, hogy milyen időintervallumban beérkezett eseményeket (naplóbejegyzéseket) kell még memóriában tárolni. Mivel a webszerver a munkamenet azonosítók érvényességét ugyanúgy idő alapján figyeli, mint ahogy a látogatókat a lekérések időkülönbsége alapján elkülönítjük, ezért adható meg a szerver beállításának ismeretében a fenti intervallum.
Mivel a további feldolgozások a felhasználói útvonalakat használják, ezért célszerű például egy adatbázisba tölteni az analizálás során keletkező útvonalakat. Így a további modellkészítési lépések is gyorsíthatóak az adatbázis kezelő rendszerek optimalizálásával.

XML alapú konfiguráció

A fejlesztés során egyértelművé vált, hogy az eddig elterjedt konfigurációs fájl struktúra nem alkalmas akármilyen beállítás tárolására. Főleg a szöveges és speciális karakterek, tömbök kezelésre nem alkalmas.
Mivel az egyes eseménynapló fájlok szerkezetileg egyediek is lehetnek, ezért lehetőséget kell biztosítani a reguláris kifejezések eseménynaplóhoz kötéséhez (vagyis a konfigurációs fájlban kell tudni megadni).
A *XML* alapú konfiguráció további előnye a csoportosíthatóság több mint két szint mélységben is. Így kényelmesebbé tehető a tömb jellegű adatstruktúrák megadása, pontosabban leírható már a címkékkel is az egyes elemek jelentése és célja.

Az *XML* fájlok feldolgozása még nem teljesen kiforrott a *PHP* esetében, de már vannak használható és gyorsan alkalmazható megoldások, kiegészítések is, amelyek tudása olykor még korlátozott, vagy egyedi fejlesztést igényel.

Kereső robot a referer alapú vizsgálathoz

A *referer* alapú vizsgálathoz szükséges, hogy az adott portálra vonatkozóan az egyes weboldalak között az összes átjárási lehetőséget megadjuk. Nagyobb, rendszeresen bővülő portálokon (például áruház, hírmagazin) ennek még az időszakosan történő megadása is igen komoly munkát igényel.
A *referer* alapú vizsgálathoz kifejleszthető egy (általánosan is használható) úgynevezett ***crawler (kereső robot)***, amely automatikusan végigjárja a portál összes oldalát, kigyűjtve az összes linket, rendszerezve az átjárási lehetőségeket. Jelenleg elterjedt portál navigációs és hirdetési jellegzetesség, hogy minden weboldal (funkció) elérhető minden weboldalról, vagy a linkek dinamikusan (adott tartalomhoz, hírhez kapcsolódóan, napszaktól függően, vagy véletlenszerűen) látogatásonként változóan kerülnek ki egy-egy weboldalra. Fentiek miatt a kereső robot igazán csak olyan esetekben tud jól segítségünkre lenni az átjárási lehetőségek begyűjtésében, ahol a portál szerkezete viszonylag fix, jól elválasztott navigációs struktúrák vannak.

XML és grafikus kimenetek

Az elkészült analizáló szoftver jelen formájában egy prototípusnak megfelelő kimenetet produkál, amelynek megértése erősen feltételezi a szoftver és a modellek ismeretét.
A modellek további feldolgozására (például egyedi megjelenítés, beillesztés más programba) ma az a legkézenfekvőbb és legjobb kompatibilitási megoldás, ha az eredményeket előre definiált szerkezetű *XML* fájlba is ki lehet menteni.
A megértést és szemléltetést könnyítheti meg, ha az eredmények grafikus formában jelennek meg (például oszlopdiagramm, gráf, folyam). Ez segítheti a nem szakértőket a szoftver mélyebb ismerete nélkül is a megfelelő döntés (szerkezet, navigáció, tartalom, stb. módosítása) meghozatalában, az analizálás eredményeinek és jelentőségének felismerésében.

Összefoglalás

Jelen cikkben egy konkrét modellező szoftver működését vizsgáltuk meg, külön kitérve az egyes algoritmikus kérdésekre.
Következő cikkemben a most bemutatottak alapján néhány konkrét példán keresztül szemléltetem, hogy mik a felhasználói viselkedés vizsgálatának gyakorlati eredményei.

Beszédes Balázs (beszedes@ei.hu)
24 éves, az e-Média Informatikánál mérnök-informatikus. Hobbija a kerékpározás és a kirándulás.