Műhely

Webhelyek archiválási problémái

DRÓTOS László – VISKY Ákos László

A cikk, amely rövid összefoglalót ad a webhelyek archiválási problémáiról, részben az OSZK Webarchívum tapasztalataira és a MIA Wiki tudásbázisra, részben pedig a Könyvtári Intézet Az internet archiválása mint közgyűjteményi feladat *című akkreditált tanfolyamához készült tananyagban összegyűjtött példákra épül.*[1]

Bevezető

Jelen cikk *Perger Ádám*nak a 2021-es *„404 Not Found – Ki őrzi meg az internetet?”* című konferencián elhangzott előadását[2] kiegészítő, általános webarchiválási problémákkal foglalkozó rész írott változata, melyben olyan tipikus problémákat mutatunk be, amelyekkel az Országos Széchényi Könyvtár (OSZK) Webarchívumának eddigi ötéves működése során gyakran találkoztunk. Ezek egy része az archiváló eszköz paraméterezésével vagy alternatív technológia használatával orvosolható, más része viszont a tartalomszolgáltató közreműködését igényli. Írásunk célja, hogy felhívja a figyelmet arra, hogyan lehet robot- és archívumbaráttá alakítani a webhelyeket, hasonlóan ahhoz, ahogy az optimalizálás az akadálymentesítés esetében történik.

Mi kell a megőrzéshez a webarchívum részéről?

Ha igaznak fogadjuk el azt az állítást, hogy egy átlagos mai honlap komplex multimédiás tartalom, akkor a mentése során szembe kell néznünk ennek a következményeivel: azzal, hogy ha megfelelő módon szeretnénk archiválni, akkor törekednünk kell ennek a komplexitásnak a megőrzésére. Persze van létjogosultsága más szempontú megközelítésnek is, például ha kutatási célból csak a szövegre van szükségünk, vagy technikai okokból kevesebbel kell beérnünk,

de ha mint nemzeti könyvtár végezzük a magyar vonatkozású tartalom begyűjtését, akkor a teljességre kell törekednünk.

Az OSZK Webarchívuma is ezt az elvet tartja szem előtt, azonban azt is tudomásul kell vennünk, hogy csak törekedhetünk a teljes körű archiválásra, mivel a jelenleg rendelkezésre álló technológiákkal szinte csak töredékeket tudunk megőrizni az interneten található tartalmakból. Egy webhely mentése sosem egyenlő a dokumentumkezelésnél megismert *save as* paranccsal, mert nem lehet bitről bitre egyező másolatot készíteni, sem automatikus, sem pedig egyedi, kézi szoftverek használatával, mivel olyan bonyolult és folyamatosan változó technológia áll egy-egy webhely mögött, amelyet a kifejlesztett eszközökkel nem mindig tudunk lekövetni. Gondoljunk csak arra, hogy az automatikus archiválás egyik fő szoftvere, a *Heritrix*[3] még a Web 1.0 idején született a statikus linkek követésére, viszont ma már a felhasználó aktivitása szerint betöltődő tartalmakról és a monitoron egységes képpé összeálló szoftvergyűjteményről beszélhetünk a honlapok esetében is – és akkor még nem is érintettük a mindennapjainkban egyre nagyobb szerepet betöltő webkettes tartalmak problematikáját. Eleve érdekes kérdés, hogy egy webhelynek melyik az autentikus változata, és ha van ilyen, akkor csak azt vagy az egyéb megjelenési formákat is archiválnunk kell-e. (A nyelvi változatok vagy az akadálymentes felület esetében természetesen ez nem vetődik fel.) Azonban például egy híroldalt ma már különféle böngészőkön keresztül is megnézhetünk, különböző operációs rendszerű számítógépeken vagy akár mobileszközökön is, de az utóbbi platformokra optimalizált változat, applikáció is készülhet, így ha szöveges eltérés nincs is, de az egyszerűsített nézetből és elrendezésből adódhatnak különbségek. Vajon a különböző böngészőkben ugyanazt és ugyanúgy látjuk? Amikor a linuxos szerveren elindul az archiváló robot, akkor tulajdonképpen mi is történik, melyik változatot mentjük? Ha elindítjuk a Heritrixet, lementi nekünk az iOS-re és Androidra készült változatokat is? Az applikációknak van egyáltalán olyan URL-je, amelyen el tud indulni valamilyen robot?

Ha úgy tetszik, hibaforrást, vagyis a mentés minőségét befolyásoló tényezőt jelentenek az archiváló eszköz képességei, az archiválandó webhely által használt technológiai megoldások, továbbá az archiváláshoz alkalmazott paraméterek is. Utóbbit magyarázza, hogy az archiválás során a webkurátor határozza meg, hogy mit szeretne lementeni (pl. egy webhely teljességét vagy csak bizonyos részeket), milyen beállításokkal dolgozzon a mentést automatikusan végző szoftver (pl. mekkora futásidő alatt, milyen mélységben, hány ugrással követheti a linkeket, és milyen fájltípusokat tölthet le), vagy a személyes közreműködéssel történő mentés pontosan mire térjen ki. Bár első pillantásra jó ötletnek tűnik, hogy egy webhelyről mentsünk le mindent, a valóságban jellemzően erre csupán alkalmi döntést követően teszünk kísérletet, és inkább csak az egyedi mentéseknél, mivel a tömeges archiválásra használt eszköz „vakon” dolgozik, ezért korlátozások nélküli „szabadon engedése” túl sok haszontalan tartalom letöltésével járna, amit senki sem győzne tárhellyel. A webarchiváláshoz rendelkezésre álló eszközök adottak, de folyamatosan törekszünk a legújabb, esetleg előrelépést biztosító technológiák használatára, illetve paraméterezhetjük őket úgy, hogy minél több releváns tartalmat be tudjunk gyűjteni. Ám mindennek sikerességét befolyásolja a szóban forgó webhely technológiája is, amelyre csak közvetett ráhatásunk van. Ez persze nagyon fontos még így is, ezért is fogalmazzák meg a webarchívumok ajánlásaikat, hogyan lehet egy webhely archívum- vagy éppen robotbarát – nem ugyanarról van szó, de mindkettőnek megvannak a maga feltételelei, amelyek segítik teljesülését.

Mi kell a megőrzéshez a tartalomszolgáltató részéről?

Robotbarátnak[4] akkor nevezünk egy webhelyet, ha annak releváns tartalma könnyen bejárható az archiváló szoftver számára, míg az érdektelen (pl. naplófájlok, segédállomány) vagy lementhetetlen részei (pl. adatbázisok, webáruházak) el vannak rejtve a szoftver elől.

***Egy robotbarát webhely főbb tulajdonságai**:*

- van honlaptérképe, amely minden lényeges aloldalra elvezeti az eszközt;
- a tartalom értékes része nincs túl mélyen a kezdőlapról indulva, és linkeken keresztül is elérhető, nem csak egy kereső űrlapon át;
- szabályos HTML-linkek vannak a *Javascript*-, *Flash*-, *Java*-alapú megoldások mellett, amelyeket a robot is követni tud;
- kerüli az azonos tartalomra mutató belső vagy a végtelen körben egymásra hivatkozó linkeket, helyette kanonizálja azokat a robotok számára;
- kerüli a *frame*-eket, az egérkattintásra aktiválódó *layer*eket, a dinamikusan generálódó tartal-

makat, inkább statikus és önálló URL-címekkel rendelkező lehetőségeket generál ezekből a robotok számára;
– rendelkezik jól konfigurált *robots.txt*-vel, amely beengedi a robotokat, de csak a tényleges tartalmat szolgáltató, számukra optimalizált részekre.

Szerencsére ezek a megoldások a kereskedelmi robotok működését is segítik, így egy webhelytulajdonosnak a saját jól felfogott érdekében is érdemes őket alkalmaznia.
Az archívumbarát[5] webhely nemcsak robotbarát, hanem jó minőségben archiválható is, a lementett változat tartalmában, megjelenésében és funkcionalitásában kellően hű mása az eredetinek. Itt már olyan szempontok is fontosak, amelyek túlmutatnak egy csak szövegelemzésre kifejlesztett üzleti célú robot szükségletein.

Egy archívumbarát webhely főbb tulajdonságai:

– logikus *site*-struktúrája van, amelynek a felépítése az URL-címekben is tükröződik;
– valid HTML- és CSS-kódokat használ, ami lehetővé teszi a helyes megjelenítést a szabványokat követő böngészőkben;
– kerüli az ékezetek és a speciális karakterek használatát az alkönyvtárak és a fájlok neveiben;
– lehetőleg nyílt fájlformátumokat használ, amelyek hosszú távon is megjeleníthetők maradnak;
– nincs benne olyan speciális formátumú tartalom, amelyhez külön megjelenítő- vagy böngészőkiegészítőt kell telepíteni;
– a hang- és a videótartalom nem csupán sugárzott (*stream*) módon van beágyazva, hanem letölthető fájlok formájában is;
– a *robots.txt* fájlban nincs letiltva a külalakot szabályozó CSS-fájlok letöltése;
– nem tartalmaz olyan szerveroldalon futó szkripteket, programokat vagy adatbázist, amely nélkül a *website* használhatatlan;
– a webszerver nem használ olyan *session* vagy *persistent* típusú *cookie*-kat, amelyek alapvetően befolyásolják a megjelenő tartalmat;
– részletes, beágyazott metaadatok vannak a weboldalak fejlécében és az egyéb dokumentumokban (pl. képek, PDF-fájlok), melyek megkönnyítik a begyűjtött digitális objektumok beazonosítását és automatikus metaadatolását;
– feltünteti a készítés vagy az utolsó módosítás dátumát a weboldalakon és a dokumentumokban, hogy az archivált változat használója meg tudja állapítani, mikor készültek, és ne csak azt lássa, hogy mikor lettek archiválva;
– jogi közleményében kitér az archiválásra (pl. „archiválható, de csak fél év után szolgáltatható, és csak könyvtáron belül"), vagy egy *Creative Commons*-licenccel szabályozza a felhasználást az archivált példány esetében is.

A minőség-ellenőrzés fontossága

A fenti felsorolásból jól látszik, hogy nemcsak a weboldalak komplexek, de azok a szempontok is, amelyeknek meg kell felelniük ahhoz, hogy archívumbarátnak tekinthessük őket. Emellett természetesen a tulajdonosnak is lehetnek szempontjai, ahogy egy fejlesztőnek is preferált megoldásai, netán éppen aktuális piaci trendeknek is meg kell felelnie; és mindezeken túl a következetlen megoldások is hibákat okoznak. Archiválási oldalról lehet és kell is jelezni a webarchiválási szempontokat a webhelyek tulajdonosai és a fejlesztők felé, lehet kérni bizonyos változtatásokat (pl. a *robots.txt*-ben, a használt technológiában), de az ajánlások vagy kérések megfogadása, teljesítése sok mindenen múlhat. Köztes megoldásként szóba jöhet valamiféle egyszerűsített adatcsere-protokoll használata, netán a tartalom csomagban történő beadása az archívumba, de ilyen esetekben számolni kell bizonyos mértékű veszteséggel (például a külalak és a funkcionalitás elvesztésével) vagy felhasználási nehézségekkel (a csomagolt és tömörített tartalom önmagában nem visszanézhető). Amivel vissza is kanyarodtunk a bevezetőben említett hibalehetőségek széles tárházához.
Ahogy a „bitpontosság" problematikájához is. Az eddigiekből talán már látható, hogy miért nem beszélhetünk bitszintű pontosságról a webarchiválás kapcsán; a „pontatlanság" mértékét azonban csak különböző ellenőrzésekkel tudjuk megállapítani, például a keletkező technikai adatok elemzésével (mely linkeket nem tudtunk követni, mekkora a lementett fájlok mérete, és milyen az összetételük) vagy a mentés visszanézésével (külalak összehasonlítása, kritikus megoldások, hiányok keresése). Ez élő munkát kívánó tevékenység, ráadásul az automatizált statisztikai adatok kinyerésével csak korlátozott információhoz jutunk, de azt is szakembernek kell kiértékelnie.[6] Mindez azért fontos, hogy egyrészt képet kapjunk arról, mit sikerült archiválni és mit nem, másrészt ezek mentén tudjuk megállapítani, mi miatt nem sikerült valamit lementeni vagy reprodukálni, ebből mit tudunk korrigálni, és mi az, ami túlmutat

a lehetőségeinken. Az ellenőrzés során arra is gondolnunk kell, hogy bizonyos esetekben az egyébként sikeresen lementett tartalom valamilyen oknál fogva nem jelenik meg a *wayback*-programokban, mert mondjuk a linket nem sikerült átírni lokálisra (pl. egy ékezetes fájlnév miatt), vagy az eredeti szerver olyan *cookie*-t vagy *session*-azonosítót használ, amely az archívumban már lejárt, vagy mivel bizonyos tartalmak (pl. videók) lejátszása az eredeti szerveren futó programokhoz van kötve.

Tipikus problémák, hibajelenségek

Az egyik leggyakoribb hibajelenség, hogy vagy semmit sem sikerül letölteni az adott webhelyről, vagy bizonyos fontos fájlok hiányoznak a mentésből. Az előbbinek két fő oka lehet: elérhetetlen az adott webhely (pl. szerverhiba vagy végleges megszűnés miatt), vagy ki vannak róla tiltva a robotok, esetleg csak bizonyos típusú böngészőket szolgál ki a webszerver. Sajnos a *robots.txt* (amelyben a robotok működését szabályozza egy adott webhely) hiányát is tiltásként értékeli a Heritrix, így el sem indul az adott helyen. A különböző okokra eltérő módon tud reagálni az archiváló munkatárs. Bizonyos idő elteltével ellenőrizheti a webhely elérhetőségét, tiltás esetén a webmestertől kérheti a robot beengedését, a *robots.txt* megváltoztatását, esetleg módosíthatja a *crawler*[7] „álcaruháját", hogy más típusú böngészőnek adja ki magát. Ha csak bizonyos elemek (pl. a külalakot beállító CSS-fájl vagy egyes beágyazott képek, videók) hiányoznak, annak is több magyarázata lehet. Okozhatja ezt is a *robots.txt*, mert például bizonyos CMS-rendszerekben a *.css* kiterjesztésű fájlok alkönyvtára eleve ki van tiltva, hiszen a *Google* és más keresőgépek számára ezek érdektelenek; vagy pedig az aratásnál túlságosan szigorúan állítottuk be a paramétereket (méret, mélység, bejárható tartomány), így a robot nem jutott el addig, ahonnan letölthette volna a szóban forgó elemeket. Ez esetben a beállítások módosításával, esetleg további *seed*-URL-ek hozzáadásával orvosolhatjuk a problémát.

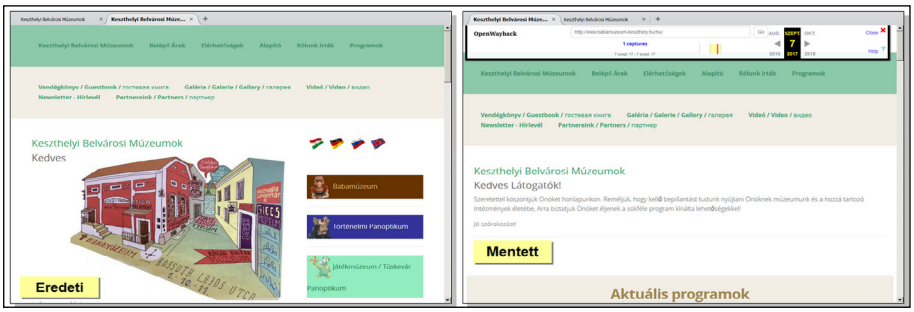
Főleg a közösségi médiában (pl. *Facebook*, *Twitter*, blogok) és a hírportálokon gyakori megoldás, hogy az oldal felső része alatti, illetve régebbi tartalmak csak olyankor töltődnek be a szerverről, ha a felhasználó lejjebb görget, amit a Heritix-szerű robotok nem tudnak imitálni, ezért csak az oldal tetején levő szövegeket, képeket mentik le. Ilyenkor érdemes egy olyan archiváló eszközt választani, amelyben van *autoscroll* (automatikus lapozás) funkció, vagy böngészőmotort használ a letöltéshez.

Végül, de nem utolsósorban sokszor azért hiányoznak egyes elemek az archivált webanyagból, mert az azokra hivatkozó linkeket olyan programok állítják elő, vagy olyan fájlokban találhatók, amelyeket a robotunk nem értelmez. Zárásképpen említsük meg azt a problémát is, hogy az elavuló technológiák vagy formátumok is hibát okoznak, sőt, akár a korábban sikeresen archivált változat is „elromolhat", mert az azt használó megoldások nem jelennek meg a böngészőkben. A weben is megfigyelhető, hogy technikai divatok jönnek-mennek és üzleti konkurenciaharc zajlik, aminek az a következménye, hogy olyan megoldásokat használnak, melyeket nem minden böngésző támogat egyformán, és amelyek gyakran biztonsági okok miatt hosszú távon nem fenntarthatók. Ilyen volt például az interaktív médiatartalmak szolgáltatására kitalált *Silverlight*, melyet a Microsoft fejlesztett, az *Adobe* cég *Flash* formátuma vagy az *Oracle Java* megoldása. Ezek hosszú távú szolgáltatása vagy a korabeli böngészők virtuális gépen való futtatásával (pl. *Oldweb.Today*), vagy pedig a megjelenítő funkcióik mai böngészőkben való emulálásával (pl. *Ruffle Flash Player*) oldható meg. Pozitív ellenpéldaként említhető maga a HTML nyelv, amelyet egy közösség fejleszt és nyílt forráskódú, ezáltal rugalmasabb és biztonságosabb.

Mindezek miatt fontos lenne, hogy a fejlesztők gondoljanak az archiválási szempontokra, legalább alternatívaként használjanak robot- és archívumbarát nyílt megoldásokat, és működjenek együtt a memóriaintézményekkel is. Erre szerette volna felhívni a figyelmet ez a rövid cikk is.

Képmelléklet

Hiba-és problématípusok
(Forrás: OSZK Webarchívum)



1. képpár

Valamiért nem sikerül reprodukálni az archivált oldal külalakját, az egyes elemek átrendeződve jelennek meg.

2. képpár

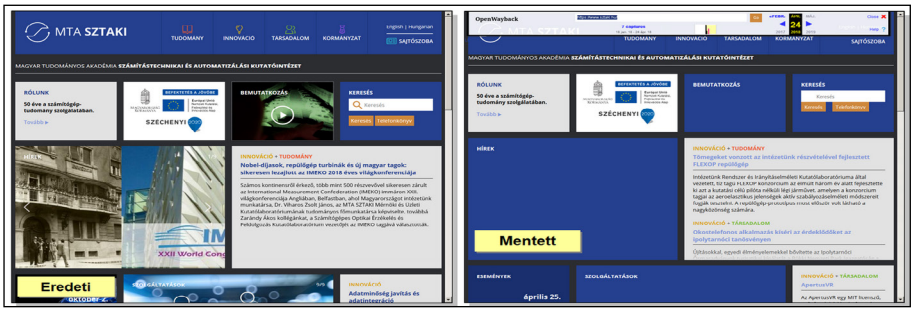
Csak a szöveges tartalom maradt meg, a külalakot meghatározó stílusfájl hiányzik.

3. képpár

A webhely megoldásai miatt a kezdőoldalról a menüpontok az akadálymentes felületre visznek.

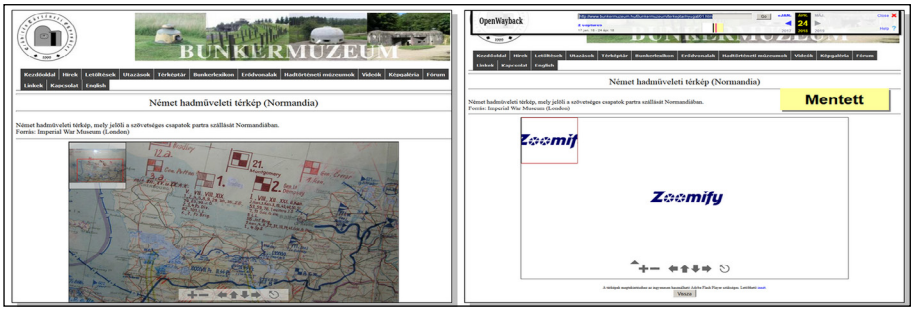
4. képpár

Az oldal tetején lévő képek kivételével a csak görgetésre letöltődő fotók nem kerültek archiválásra.



5. képpár

Az állóképeket vetítő és a videókat lejátszó modulok nem működnek a kezdőlapon.

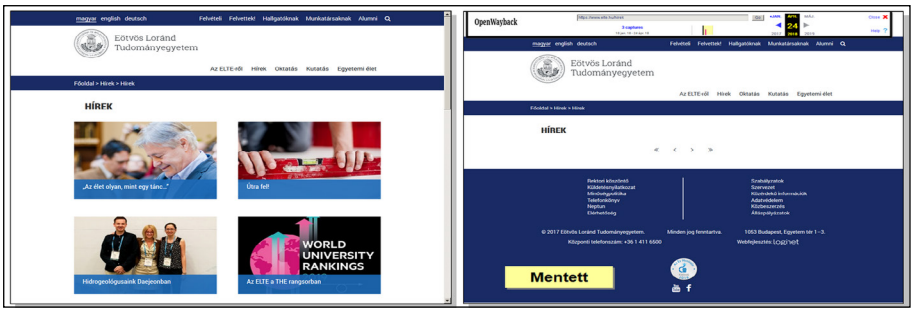


6. képpár

A térkép csak egy interaktív Flash-alapú megjelenítővel nézhető meg, amelyet se archiválni, se a mai böngészőkkel visszanézni nem lehet.

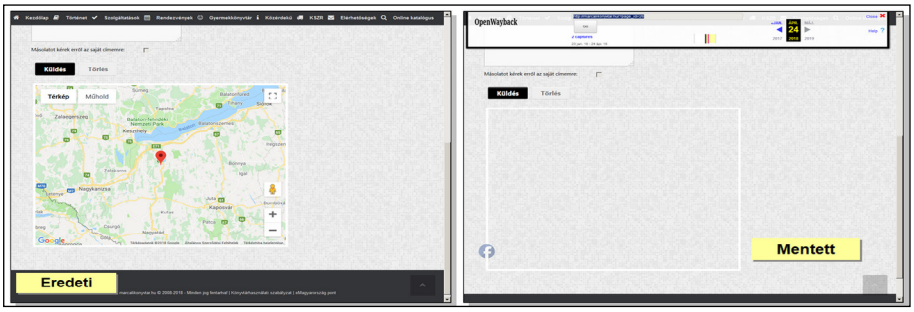
7. képpár

A robots.txt-ben lévő tiltás miatt az archiváló szoftver semmit nem tudott lementeni.



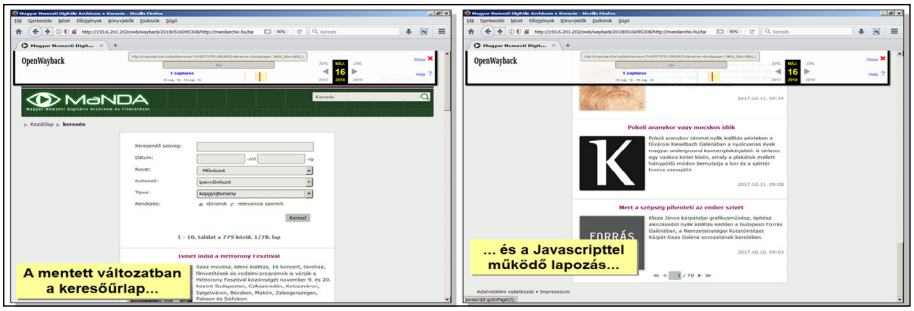
8. képpár

A hírek Ajax scripttekkel vannak beágyazva a weboldalba, amelyeket a robot nem tudott végrehajtani, ezért ezek hiányoznak.



9. képpár

A külső szerverről beágyazott GoogleMaps és a Facebook-widget nem jelenik meg.



10. képpár

Nem működő belső kereső a háttéradatbázis hiánya miatt és nem működő lapozónyilak a javascriptes megoldás miatt.

Jegyzetek és irodalmi hivatkozások

1. OSZK Webarchívum – Az Országos Széchényi Könyvtár webarchiváló projektjének honlapja. MIA Wiki – tudásbázis aloldal. Budapest: OSZK, cop. 2022. Hozzáférhető: https://webarchivum.oszk.hu/szakembereknek/mia-wiki/ [Megtekintve: 2021.11.30.]; OSZK Webarchívum – Az Országos Széchényi Könyvtár webarchiváló projektjének honlapja. Az internet archiválása mint közgyűjteményi feladat aloldal. Budapest: OSZK, cop. 2022. Hozzáférhető: http://webarchivum.oszk.hu/honlap/tananyag/ [Megtekintve: 2021.11.30.]; Az internet archiválása mint közgyűjteményi feladat: Akkreditált tanfolyam [online]. == A Könyvtári Intézet honlapja. Tanfolyamok aloldal. Budapest: Könyvtári Intézet, cop. 2022. Hozzáférhető: https://ki.oszk.hu/tanfolyamok/az-internet-archivalasa-mint-kozgyujtemenyi-feladat [Megtekintve: 2021.11.30.]

2. Az előadás címe: A Berzsenyi Dániel Könyvtár első lépései a webaratás útján – A bdmk.hu honlap archívumbaráttá alakítása. Ld. „404 Not Found" workshop – 2021. november 23–24. [online]. == OSZK Webarchívum – Az Országos Széchényi Könyvtár webarchiváló projektjének honlapja. Szak-

embereknek – „404 Not Found" workshop – 2021. november 23-24. aloldal. Hozzáférhető: https://webarchivum.oszk.hu/szakembereknek/404-not-found-workshop/404-not-found-workshop-2021-november-23-24/ [Megtekintve: 2022.09.01.]

3. Heritrix [számítógépes szoftver]. == MIA WIKI – A Magyar Internet Archívumhoz készülő tudásbázis. Hozzáférhető: https://webarchivum.oszk.hu/mediawiki/index.php/Heritrix [Megtekintve: 2021.11.30.]

4. Crawler-friendly website. == MIA WIKI – A Magyar Internet Archívumhoz készülő tudásbázis. Hozzáférhető: https://webarchivum.oszk.hu/mediawiki/index.php/Crawler-friendly_website [Megtekintve: 2021.11.30.]

5. Archive-friendly website. == MIA WIKI – A Magyar Internet Archívumhoz készülő tudásbázis. Hozzáférhető: https://webarchivum.oszk.hu/mediawiki/index.php/Archive-friendly_website [Megtekintve: 2021.11.30.]

6. Az adatok elemzésével észrevehetjük, hogy valami történt, például jelentősen kisebb méretű lett a mentésünk, mint a korábbiak voltak, vagy sokkal több link maradt feladatban (azaz hiúsult meg a letöltés), mint máskor, de az okokra nem kapunk automatikus, kész választ; és természetesen mindez másképp jelenik meg egy gyűjtemény vagy egy webhely aratásánál. Más jellegű probléma, hogy tömeges archiválás esetén milyen mértékben tudjuk ellenőrizni az eredményt, van-e kapacitás több száz vagy akár több ezer cím átnézésére. Hasznos lehet az adott webhelyről más gyűjteményben készült mentések átnézése is, hogy mennyire sikerült elérni azt a szintet, mint amit azokban látunk. Könnyű belátni, hogy az ellenőrzésnek is különböző mértéke lehet annak függvényében, milyen minőségű mentés elérése a célunk. Ennek a tevékenységnek lehetnek még a mentést megelőző lépései is (például annak ellenőrzése, hogy milyen technológiát használ az adott webhely), de mivel ez sokszor nem perdöntő, inkább az utólagos minőség-ellenőrzés a jellemző.

7. A crawlerek különféle webtartalmak teljes körű mentésére szolgáló szoftverek, ellentétben a scraperekkel, amelyek csupán a weblapok bizonyos részeit (pl. a metaadatokat, szövegeket, képeket) mentik. Ld. DRÓTOS László. Crawlerek és scraperek: Webes tartalmak mentésére szolgáló programok. == Könyv, Könyvtár, Könyvtáros [online], 30. (2021) 9., p. 18–28. HU ISSN 1216-6804 (Nyomtatott), HU ISSN 2732-0375 (Online). Hozzáférhető: http://ojs.elte.hu/3k/article/view/3393/ [Megtekintve: 2021.11.30.]

(Beérkezett: 2022. március 30.)