

URL-menedzselés a könyvtárban

A hálózati digitális gyűjtemények kezelése, szolgáltatásként való megjelenítése nem kis feladat elé állítja a könyvtárakat. A digitális tartalmakhoz való hozzáférés lényeges pontja a *link (ugrópont)* gondozása – a link sikeres működése pedig az egységes forráslokalizátor, az *URL (uniform resource locator)* függvénye. A weblapú tartalom lehet szabad hozzáférésű és előfizetett (licenctíjas), helyi vagy távoli, illetve az OPAC-ból közvetlenül meghívható, vagy a könyvtár weblapjához kapcsolt. Bár az URL-gondozás a mai könyvtárosság egyik időigényes és kényes feladata, meglepő módon szegényes a szakirodalma. A régi könyvtári rendszerben a cédulakatalógus egy tétele – de még az online világban az OPAC egy rekordja is – pusztán reprezentációja a fizikai állománynak, nincs közöttük közvetlen elérés. A használó megtalálja a tételt, kiírja az elhelyezés adatait, a raktári jelzetet, és annak alapján kikéri, vagy szabadpolcon megkeresi a dokumentumot.

Az online világból érkező forrásokkal megváltozik a helyzet: a könyvtár átjáróvá, kapuvá válik, s az URL révén juttat el az információ forrásához. Ilyen funkciót lát el a könyvtár a vendorok közötti linkfeloldókkal (resolver), pl. az új szabványra, az OpenURL-re épülő SFX, a CrossRef és a LinkSource esetében, amelyek különböző szolgáltatók által közzétett anyagokat közvetítenek, illetve az egy vendorhoz kapcsolódó rendszereket kötik össze (pl. az Ovid, OCLC rendszereit). *Laura Cohen* írása nem öleli föl a vendorok által felügyelt linkkapcsolatokat, hanem csak azokkal az URL-ekkel foglalkozik, amelyek a könyvtáros ellenőrzése alatt állnak. Az URL-menedzselésben a könyvtáros számára három kulcsfeladat van. Az abszolút és relatív URL-ek gondozása, a könyvtárstruktúra indexéhez vezető ún. könyvtár-URL-ek létesítése, valamint az ugrópontok folyamatos ellenőrzése.

Külső linkhez mindig **abszolút URL**-t kell használni. Az URL szerkezete a következő: *protokoll://hoszt.második-szintű-domén.első-szintű-domén:port/lokális-elérési-út/fájlnév* – illusztrációnak a szerző intézményének a honlapját vesszük végig a cikk során: *http://library.albany.edu:80/services/microforms.html* –, itt tehát a protokoll a *http*, a második szintű domén a *library*, az első szintű domén az *albany.edu*; a *80*-as port pedig a web kitüntetett portja, ami el is maradhat az URL-ből egyébként. A külső link, a *library.albany.edu* arra

szolgál, hogy a böngésző le tudja kérdezni a doménnévrendszertől, a DNS-től, hogy hol van az a webhely, ahol a kérdéses fájl – *microforms.html* – található. A DNS egy globális szerverrendszer, amely adott webhelyekhez irányító mutatókat tartalmaz: a hoszt- és doménneveket lefordítja a megfelelő numerikus IP-címre – pl. a *library.albany.edu* kérdésre a DNS-rendszer a *169.226.11.130* IP-címet adja ki.

A belső hálózatra vonatkozó linkeket **relatív URL** segítségével célszerű megadni. A relatív URL a helyi rendszer címstruktúráján belül más ponton elhelyezkedő fájlra mutat. Például a *services/microforms.html* olyan fájlt képvisel, amely a linkelő fájl alatti szinten van, a */services/microforms.html* pedig egy szinttel feljebb lévő fájlra mutat. A relatív URL egyik változata az **abszolút lokális URL**, amely a helyi weblap gyökérkönyvtárából indul ki: */services/microforms.html*. A kezdő perjel jelzi a gyökérkönyvtárat – jelen esetben a *services* mappa a gyökérkönyvtárban van. Ez az URL-konvenció nagyon praktikus és hasznos, mert a kezdő perjel alapján a webszerver megtalálja a gyökérkönyvtárat, és bárhova is helyezzük a linkelt fájlt a webhelyen belül, a link működni fog!

A gyökérkönyvtár alatti mozgathatóság praktikus-sága mellett a relatív URL másik fontos tulajdonsága, hogy ugyanazon szájon belül egy fájlhoz történő linkelés nem igényli egy külső felderítő eszközt, a DNS alkalmazását. Ha a belső linkeket mind relatív URL-lel készítjük el, a DNS-hez futó főlegesen forgalom kiiktatódik. A kutatás kimutatta, hogy a DNS-forgalom csökkenése határozottan javíthatja az elérési sebességet.

A könyvtár-URL

Ez az URL-típus – angolul directory URL – fájl nélkül van megadva; csak a könyvtárszintig, a fájlstruktúra tetejéig vezet, pl. *http://library.albany.edu*. Ekkor az URL lényegében az *index.html* fájlra mutat – ennek a fájlnak a nevét nem kell megadni, mert a webszerver defaultként értelmezi. Be lehet állítani a webszervert, hogy egy vagy több fájlnevet alapértelmezésként kezeljen. A Unix alapú gépeknél az alapértelmezésű fájl az *index.html*, Windows alapú szervereknél ez lehet *index.htm* vagy *default.htm*.

A directory URL a webhely fő pontjára vezet, vagy legalábbis egy csomóponthoz. Nincs nagy kockázat, ha nem adunk meg default fájlnevet, mivel a webszerver bármely egyéb konstruált új fájl névre működni fog tovább. De nem is kell megadni, jobb, ha így elrejtve működik. Ezek a fájlnevek nem is adnak felvilágosítást tartalmukról, láthatóságuk emiatt nem jelent előnyt. A tartalomról szóló szemléletes nevet webfájlokhoz érdemes adni.

Amikor az alapértelmezésű fájl név nincs megadva a directory URL-ben, az URL végére egy perjelet kell tenni, pl. *services/*. A záró perjel jelzi a szervernek, hogy a böngésző egy olyan fájlt kér, amely az adott szerveren alapbeállításként szerepel. Ez az operáció két tranzakcióban testesül meg. A böngésző meghívja a szervertől a default fájlt, majd a szerver azonosítja, és elküldi a böngészőnek. Ha a perjel elmarad, a szerver és a böngésző közötti forgalom megduplázódik: két tranzakcióból négy lesz:

1. A böngésző lekéri a szervertől a *library.albany.edu/services* fájlt.
2. A szerver keresi a *services* fájlnevet a gyökérkönyvtárban, majd válaszol a böngészőnek, hogy nem találja.
3. A böngésző kéri a szervertől, hogy keresse meg a *services* nevű alkönyvtárat, és abban az alkönyvtárban egy olyan fájlt, amely defaultként van konfigurálva.
4. A szerver másodszor is keresést végez a gyökérkönyvtárban, megtalálja a *services* alkönyvtárat és a default fájlt (*index.html*), és elküldi a fájlt a böngészőnek.

A kevesebb tranzakció a böngésző és a szerver között csökkenti a szerver megterhelését, és csökkenti a hálózati forgalmat. Ez pedig gyorsabb válaszidőket jelent a böngésző kérésére. A záró perjel mindenképpen csökkenti a szerverre nehezedő forgalmat.

Linkellenőrzés

A linkellenőrzés az URL-gondozás fontos eleme. A holt linkek csökkentik az elérési eszköz megbízhatóságát, és frusztrálják a használatot. A linkeket legalább havonta ellenőrizni kell. A szabad forrásokhoz mutató halott linkek kiszűrése nagyon munkaigényes tud lenni, hiszen rengeteg link válik holttá naponta. A könyvtárosoknak meg kell küzdeniük ezzel a tűnékenységgel, meg kell oldani a törött linkek áttételét külön fájlba; ki kell komment-

tálni a törött linkeket ellenőrzésig; és amikor nincs bizonyíték az élő linkre, akkor ki kell iktatni.

Előfizetési digitális forrásokra mutató URL-ek gondozása

A licenc alapján elérhető tartalomhoz mutató linkekkel jellegzetesen könyvtárakban és információszolgáltatóknál kell foglalkozni. Ez a terület elég új a könyvtárosok számára. A licenc alapú források sokféle vendort feltételeznek, és a szabad webforrásokétól eltérő URL formával rendelkeznek.

Kezdőoldali URL-ek készítése

Amikor egy könyvtár új digitális tartalomra fizet elő, első lépésként az adott forráshoz vezető kiindulóponti URL-t kell meghatározni. A kiindulóponti vagy kezdőoldali URL egy OPAC-mezőben, weboldalon vagy adatbázisrekordban van rögzítve, amely a böngészőt összeköti a kívánt webforrással. A kezdőoldali URL eltérhet a cél-URL-től. Kezdőoldali URL készítése általában akkor lesz aktuális, amikor értesítést kapunk a vendortól, hogy az elérést aktivizálták. Ekkor vagy megadnak egy URL-t, vagy megadják az URL helyi konfigurálásának lépéseit. Persze az értesítés el is maradhat: ez olyankor történik, ha a könyvtár a nyomtatott anyag előfizetése jogán kap elérési lehetőséget az online változathoz. Ekkor előfordulhat, hogy a könyvtárosnak kell a kezdőoldali URL-t megalkotni. Célszerű ilyenkor tüzetesen átnézni a vendor honlapját, és a vonatkozó utasításokat kiszűrni onnan, illetve a vendor segítségét kérni.

A vendor által megadott kezdőoldali URL-t természetesen a könyvtáros teszteli, mielőtt nyilvánosan elérhetővé teszi. Sokszor előfordul azonban, hogy az ajánlott URL módosításra szorul – pl. hanyagul nem tartalmazza a záró perjelet. Azt általában pótolni kell, de azután ismét ellenőrizendő, hogy a megfelelő kapcsolat létrejön-e. Ha az URL statikus, a cél-URL ugyanaz, mint a kezdőoldali URL. Ha az URL egy szkriptet hív meg a kapcsolat felépítéséhez, azaz ha az URL dinamikus, akkor a perjelet csak akkor tegyük a végére, ha a szkript az alapértelmezés szerinti fájl névről (pl. *index.cfm*) fut. Legtöbbször azonban egy fájl név, amely egy szkriptet hív meg, nem lesz szervertől alapértelmezésnek beállítva. A vendor által megadott URL-en belül látható szkriphívások (pl. *http://www3.interscience.wiley.com/cgi-bin/jtoc?ID=32356*) esetében a záró perjel megadása értelemszerűen nem is kerül szóba.

Átkattintó URL és formula-URL

Egyszerű dolog, amikor kezdőoldali URL-t úgy hozunk létre, hogy a vendor honlapján sorozatos kattintásokkal jutunk el a céloldalhoz. Az ily módon derivált URL-t átkattintó URL-nek hívhatjuk. Ezek mindig azonosak a kezdőoldali URL-lel. Az átkattintó URL-ek jellemzője az állandóság: egy forráshoz kapcsolnak, amikor egy böngésző kéri az URL-t. Az ilyen URL-ek a weben nagyon elterjedtek.

A könyvtáros azonban nem lehet biztos abban, hogy ha keresztül kattint a vendor honlapján a célponti oldalhoz, az ott talált URL állandó. Számos vendor kifejezett kívánsága, hogy az előfizető könyvtár formalap alapján készítse el a kezdőoldali URL-t – az online kapcsolat létrejötte sokszor csak ezzel az URL formulával lehetséges. A kötelező URL formula némelyike átkattintó URL, tehát állandó, míg mások átalakulnak a szkriptek meghívásakor.

Az OCLC volt az első adatbázis-forgalmazó, amely a FirstSearch beindításakor megkövetelte az URL formula használatát, majd követte a SilverPlatter, az Ovid, a ProQuest és a Cambridge Scientific Abstracts is. Számos e-folyóirat-forgalmazó, -szolgáltató és adatbázis-aggregátor finomította a formulastratégiát olyan szintaxis felajánlásával, amely folyóirat-címoldalokhoz, egyes számokhoz, archivált számokhoz, cikk-kivonatokhoz és adott cikkekhez köt. Ezt mélylinkelésnek (deep linking) hívják. A BioOne a periodika ISSN-számát használja régi folyóiratszámokhoz mutató mélylinkelésre, az EBSCO Academic Search pedig megengedi, hogy állandó linket (Persistent Link) létesítsünk a cikkek szintjén.

Az URL formuláknak lehet egyedi céljuk. Az Ingenta szükségesnek tartja, hogy a *www*-t mint hosztnevet mindegyik kezdőoldali URL-jébe tegyünk be. Amikor a böngésző egy kérést küld el, a hosztot dinamikusan menet közben jelöli ki a rendszer a kérő böngésző földrajzi elhelyezkedése, és a hívás idején meglévő hálózati forgalom függvényében. Egy *www.ingentaselect.com/* kezdetű URL *rosina.ingentaselect.com/*, illetve *pippo.ingentaselect.com/* stb. formájúvá válhat a hosztok sessionről sessionre való váltogatásától függően.

A vendorok zöme eléggé hanyagul – sokszor abszolúte tévesen – adja meg a kapcsolódási utasításokat honlapján. A Haworth Press pl. nem is adja meg a kezdőoldali URL-t. Honlapján egy menü

van, amely linkeket kínál a könyvtár előfizetett folyóirataihoz. Amikor egy linket kiválasztunk, egy JavaScript program lép működésbe, és egy új böngésző ablakot nyit meg a kért folyóirathoz való kapcsolódáshoz. Ez az új ablak azonban nem tartalmazza a böngésző címsorát, amely az URL-t kiírta. A webcím eléréséhez a windowsos felületnél a könyvtárosnak a jobb egérgombbal ki kell választania a Properties (Tulajdonságok) menüpontot, és onnan kell bemásolni az URL-címet.

A linkelési utasítások nem pontosak, nem frissek. Pl. a ma már az Ovid égisze alatt futó SilverPlatter nem tartotta karban az adatbázis családnévlístáját, sőt az egyesülés után sokáig nem volt elérhető az Ovid honlapján. Nem adtak tájékoztatást szintaxisajánlásuk következményeiről sem. Pl. egy olyan URL-t ajánlottak, amely az MLA Bibliographyt (MLAB) kombinálta az MLA Directory of Periodicals (MLAD) címtárral. Bár az MLAB-nek van tezaurusza, az adott URL olyan adatbázishoz köti az olvasót, amelynek nincs. Ez azért van, mert a MLAD-nek nincsen tezaurusza, s ugyanazon URL alatt a családnevek kombinálása elsődlegesen a tezaurusz nélküli adatbázishoz kapcsol. A könyvtáros ezt úgy hidalja át, hogy mindegyik címre külön URL-t ad meg, de időt vesz igénybe, míg rájön a problémára.

Távoli elérés és a proxy szerver

A könyvtári fejlesztés fontos eleme mostanában a távoli (off-site) elérés hálózati forrásokhoz proxy szerveren keresztül. A proxy szerver azonosítást, jogosultság-ellenőrzést (autentikálás) igényel a belépőtől. A jogosultság-ellenőrzés követelménye rendszerint a jogosult hálózati tartományon kívül eső IP-címmel rendelkező böngészők kéréseinél jelentkezik. Amint a használó igazolja magát, a böngésző a jogosult IP-tartományon belülinek tűnik fel, és megkapja az engedélyt a korlátozott webhelyhez. Ez az eljárás azokat a licencmegállapodásokat követi, amelyek a jogosult használói kör számára korlátozzák a vendor nyújtotta tartalmat. A korlátozott tartalomelérés szabályozására ma a proxy szerver alkalmazása sokkal népszerűbb, mint a jelszavas jogosultság-ellenőrzés.

A proxy szerver konfigurációs fájljának elkészítése után a jogosultság-ellenőrzés kérése megtörténik, amikor egy távoli böngésző olyan forráshoz kér elérést, amelynek címe szerepel a fájlban. A böngésző által irányított proxy szerver, mint pl. a Netscape Proxy Server, egy proxy autokonfigurá-

ciós (pac) fájlt működtet, amelyhez a böngészők kapcsolódnak, amikor egy konfigurált forráshoz kapcsolást kérnek. A szerver-szoftver proxy, mint pl. az EZproxy is egy központi konfigurációs fájlt aknáz ki, de a böngészőnek nem szükséges arra mutatnia. Ehelyett a kezdőoldali URL-ek tartalmaznak egy előtagot, amely a konfigurációs fájlt tartalmazó proxy szerver gépre mutat. Íme egy példa: <http://silver.ulib.albany.edu/login?url=http://www.netlibrary.com/> Ez arra utasítja a böngészőt, hogy küldje a kérését a proxy szerveren keresztül, hogy a jogosultság-ellenőrzés megtörténhessen.

Egy proxy konfigurációs fájl fenntartásához elengedhetetlen a kezdőoldali URL alapos ismerete. A böngésző által irányított proxy szerverek konfigurációs fájljai a domén inputját igénylik, pl. annualreviews.org. Az EZproxy bonyolultabb elemzést kíván. A vendor kezdőoldali URL listájában minden egyes hosztnévnek külön bejegyzést kell kapnia a konfigurációs fájlban. Pl. mindegyik cím az *Annual Reviews* weboldalán külön hoszton található. Az *Annual Review of Anthropology* hosztja anthro.annualreviews.org, az *Annual Review of Microbiology* hosztja pedig a micro.annualreviews.org, amellett külön hosztleírást is igényel a konfigurációs fájlban: *Host micro.annualreviews.org*. Más hosztok és domének bevitelle függ attól, hogy a vendor száján milyen technológia van – ebben a munkában a könyvtáros minden jel szerint sok-sok „próba szerencse” eljárással fog szembesülni. Az EZproxy is egyedi leírást követel, ha a vendor JavaScriptet alkalmaz a kapcsolati folyamathoz, amint ezt a következő példa mutatja: *Domain Javascript netlibrary.com*. Összefoglalva elmondható, hogy a licencelt források webhelyen kívüli (off-site) elérése proxy szerveren keresztül újabb dimenziót és komplexitást ad hozzá a könyvtáros által vállalt URL-menedzselési munkához.

Az URL-kezelés stratégiái a könyvtárban

A könyvtári honlapok és OPAC-ok számos átfedést tartalmaznak a linkek tekintetében, tehát egy forrásra több helyről mutathat link. Az URL-kezelésnél nem banális kérdés, hogy mi történik, ha egy URL megváltozik? Holt linkhez vezethet a vendorváltás, új interfész beállítása, a vendor webhelyének átalakítása stb. Nagy terhet ró az intézményre, ha minden egyes URL-változást külön-külön mindegyik helyen korrigálni kell. A szerző szerint úgy lehet a problémát megoldani, hogy egy központi URL adatbázist hozunk létre, amelyben a

könyvtáros által fenntartott, a belinkelt forrásokhoz mutató URL-eket tartjuk. Egy ebben az adatbázisban székelő URL aztán elérhető egy átirányító, *redirect* URL-lel, amely egy szkript segítségével felkutatja a megfelelő bejegyzést, és a közvetlen URL-t küldi vissza a böngészőnek. Például <http://library.albany.edu/databases/libresre.asp?resourceid=2275> kéri a 2275. rekordot az adatbázisban, és előhossa a rekordban levő URL-t a kapcsolás létesítésére. Amikor egy URL megváltozik, a korrekciót elég egyetlen rekordban elvégezni, amely az arra mutató összes URL-t kicseréli.

A *redirect* URL-nek további funkciói lehetnek. Egy weblapot közbe lehet iktatni a tranzakció közben, ha a szkript fölfedezi, hogy a használó a jogosult webhelyen kívül van. Ez a lap távoli elérési instrukciókat tartalmazhat. Sőt, a *redirect* URL mögötti szkriptet úgy lehet megírni, hogy vegye észre, ha egy adatbázisrekord a proxy szerverhez konfigurált licencelt forrásként van jelölve. Az EZproxy esetben be lehet tenni a kívánt proxy előtagot, hogy a jogosultság-ellenőrzést megindítsa. Ennél a módszerrel a könyvtárosnak nem kell adatbevitelkor betennie az előtagot minden egyes URL-be. További előny, hogy a proxy előtag bármely későbbi módosítása könnyen kezelhető.

Javaslatok, ajánlások

Ahogy az URL-kezelés alapvető feladattá válik a könyvtárakban, az a benyomásunk támadhat, hogy ezt a munkát a könyvtáros saját szakállára, passzióból végzi. A könyvtárosok URL-eket alkotnak eredeti katalogizálási tevékenység gyanánt, ám ehhez semmiféle kalauz, útmutató (best practice) nem áll rendelkezésükre. Szükség lenne egy olyan központi gyűjtőhelyre, divatos szóval tudásbázisra, ahol részletes utasítások találhatók a kezdőoldali URL-ek megalkotásához a különböző vendorok által igényelt formában. Amolyan szakmai vita zajlik erről a témáról pl. az EZproxy elektronikus levelezőlistáján, közben a szerző maga is kísérletezik egy levelezőlista létrehozásával: <http://library.albany.edu/clearinghouse/>.

A könyvtárosoknak támogatniuk kell a mélylinkelési tevékenységet, és terjeszteniük kell a digitális tartalomhoz vezető mélylinkelés lépéseit. Ilyenek amúgy is egyre növekvő számban jelennek meg a könyvtári weboldalakon. Jelezniük kell, melyik vendor kínál tartós linkeket, és az hogyan található meg. A könyvtáraknak ki kell dolgozniuk az URL-kezelés feladatkörét a munkatársak között. A

könyvtártani oktatásban szerepet kell kapnia ennek a tárgykörnek, akár a tananyag részeként.

A könyvtár által menedzselt URL-ek száma bizonyára növekedni fog. A kereskedelmi linkfeldolgozó szolgáltatók népszerűsége ellenére a könyvtárosoknak továbbra is van és lesz feladatuk a digitális gyűjteményekhez vezető URL-ek gondozásában. A szabad webforrásokhoz vezető URL-ek gondozása

nem nélkülözheti a könyvtáros hozzáértését és szakértelmét. Ez a tevékenységcsokor a könyvtárosi szerepek további gazdagodását jelenti.

/COHEN, Laura: *Issues in URL management for digital collections.* = *Information Technology and Libraries*, 23. köt. 2. sz. 2004. p. 42–49./

(Bánhegyi Zsolt)

Az OWL Web Ontológia Nyelv – áttekintés

Az *OWL Web Ontológia Nyelv* a dokumentumokban tárolt információk gépi feldolgozására fejlesztették ki. Az OWL lehetővé teszi, hogy explicit módon ábrázoljuk egy meghatározott szókészlet kifejezéseinek jelentését, valamint ezek összefüggéseit. A fogalmak és köztük levő összefüggések ilyen ábrázolását nevezzük ontológiának. Az OWL a DAML + OIL web ontológiai nyelv továbbfejlesztett változata, amelybe beépítették a már megszerzett tapasztalatokat. Az OWL nyelvet úgy tervezték, hogy megkönnyítse a számítógépek számára a világhálón tárolt információk automatikus feldolgozását és integrálását. Ez a dokumentum a nyelv rövid bemutatására és lehetőségeinek számbavételére készült, valamint ismerteti az OWL alnyelvek jellegzetességeit és nyelvi konstrukcióit.

Az OWL három, növekvő kifejező erejű alnyelvből áll, amelyeket specifikus fejlesztői és felhasználói közösségek céljaira terveztek:

1. Az *OWL Lite* elsősorban az osztályozási hierarchiakat és egyszerű korlátozásokat alkalmazó felhasználók segítségére szolgál.
2. Az *OWL DL* (Description Logics) azokat a felhasználókat támogatja, akik a maximális kifejezőképességet igénylik a teljes számíthatóság és az eldönthetőség megmaradása mellett. Az OWL összes nyelvi konstrukcióját tartalmazza bizonyos korlátozásokkal.
3. Az *OWL Full* a maximális kifejezőerőt és az RDF szintaktikai szabadságot igénylő felhasználók számára készült. Ezért cserében azonban le kell mondaniuk a kiszámíthatósági garanciákról.

Az OWL Full az RDF kiterjesztéseként fogható fel, míg az OWL Lite és az OWL DL az RDF egy korlátozott nézetének a kiterjesztéseként. Minden OWL (Full, Lite, DL) dokumentum egyben RDF dokumentum is, és minden RDF dokumentum egyben

OWL Full dokumentum. Nem tekinthető azonban legális OWL Lite és OWL DL dokumentumnak minden RDF dokumentum, ezért óvatosságra van szükség akkor, amikor a felhasználó egy RDF dokumentumot szeretne OWL környezetbe áthelyezni.

Az OWL Lite az OWL nyelvnek csupán néhány lehetőségét alkalmazza, és több megkötést is tartalmaz. Az OWL Lite osztályai csak névvel rendelkező főosztályként definiálhatók, egy főosztály nem lehet tetszőleges kifejezés. Az OWL Lite a kardinalitás fogalmát is szűkíti, mivel a kardinalitás értéke nem lehet más, csak 0 vagy 1.

Az OWL Lite RDF sémával kapcsolatos komponensei a következők:

- *Class*: az egyedek azon csoportját határozza meg, amelyek együvé tartoznak azon az alapon, hogy egyes tulajdonságaik közösek.
- *rdfs:subClassOf*: osztályhierarchiát alkotnak, valamely osztály alosztálya egy másiknak.
- *rdf:Property*: tulajdonságok segítségével fejezzük ki a viszonyokat az egyedek között, valamint az egyedek és adatértékek között.
- *rdfs:subPropertyOf*: tulajdonsághierarchiakat szervezhetünk, valamely tulajdonság altulajdonsága a másiknak.
- *rdfs:domain*: egy tulajdonság érvényességi köre meghatározza azon egyedek körét, amelyekre az adott tulajdonság alkalmazható.
- *rdfs:range*: egy tulajdonság értéktartománya meghatározza azon egyedek körét, amelyek a tulajdonság értékei lehetnek.
- *Individual*: az Egyed fogalom az osztályok egyes példányait, előfordulásait jelenti.

A következő elemek az OWL Lite egyenlőség és különbözőség fogalmköréhez kapcsolódnak: