

Debreceni Egyetem

Informatikai Kar

Métisz

Szabadon választott könyvtár-informatikai program fejlesztése,
egy elektronikus tudományos folyóirat kezelő rendszer

Témavezető:

Eszenyiné dr. Borbély Mária
egyetemi adjunktus

Készítette:

Dudás Csaba
informatikus
könyvtáros hallgató

Debrecen

2010

Tartalomjegyzék

1	Bevezetés	6
2	A három „O”: Open Source, Open Standards, Open Access	7
2.1	Open Source – Nyílt rendszerek	7
2.2	Open Standards – Nyitott szabványok	8
2.3	Open Access – Nyílt hozzáférés	9
3	Métisz?	14
3.1	Miért saját?	14
3.2	A Métisz elődje (OJS hibák)	15
4	A php és egyéb szolgáltatások	17
4.1	PHP	17
4.2	LDAP	18
4.3	SOAP	18
4.4	A DEA.....	18
5	A Métisz maga	19
5.1	Irányelvek, gondolatok, megvalósítások	19
5.2	Könnyű laza adatszerkezet.....	20
5.3	Könnyű és egyszerű kezelés.....	20
5.4	A beállítások széles tárházát biztosítani.....	20
5.5	A munkafolyamatok elkülönítése	21
5.6	Munkafolyamatok szerkesztő általi irányítása.....	21
5.7	Rendszerfüggőségek csökkentése.....	21
5.8	A jogosultságok könnyű kezelése.....	21
5.9	A többnyelvűség megteremtése.....	22
5.10	Gyorsaság, és stabilitás	22

5.11	Rendszer folyamat rugalmassága	22
6	Függvények, csomópontok.....	24
6.1	Adatbázis függvények	24
6.1.1	lekerdezes()	24
6.1.2	letrehozás()	24
6.1.3	modosítás()	24
6.1.4	torles()	25
6.2	Jogosultsági függvények.....	25
6.2.1	jogom()	25
6.2.2	cikjoga().....	25
6.3	Megjelenítési függvények.....	26
6.3.1	linkg()	26
6.3.2	kiirg().....	26
6.3.3	hibag()	26
6.4	Log függvények	26
6.4.1	logol()	27
6.4.2	clogol()	27
6.5	Fájlkezelő függvények.....	27
6.5.1	csatolmany_fel().....	27
6.5.2	csatolmanyok().....	28
7	Kezelés és általános elvek	28
7.1	A Métisz kezelése.....	28
7.1.1	Szerző.....	28
7.1.2	Bíráló.....	31
7.1.3	Szerkesztő	33
7.1.4	Regisztráció	36

7.1.5	Profil.....	37
7.1.6	Rendszerfordítás, szó/kifejezés kezelés	37
8	Várható fejlesztések.....	39
9	Összefoglalás	40
10	Idézett forrásmunkák	41

Ábrajegyzék

1. ábra: Sikeres belépés után	28
2. ábra: Ellenőrző lista	29
3. ábra: Szerzői űrlap.....	29
4. ábra: Cikk űrlapja.....	30
5. ábra: Cikk feltöltése.....	30
6. ábra: Bíráló felülete	31
7. ábra: Bíráló felülete 2	32
8. ábra: Szerkesztői felülete cikk lista.....	33
9. ábra: Cikk elküldése	34
10. ábra:Személy kiválasztása.....	34
11. ábra: Felkérés elküldése	35
12. ábra: Cikk útja.....	35
13. ábra: Cikk adatai.....	36
14. ábra: Regisztrációs űrlap	37
15. ábra: Fordító felület	38

1 Bevezetés

Dolgozatom első fejezetében azokat a nemzetközi elveket és gondolatokat illetve kezdeményezéseket mutattam be, amelyek lehetőséget adnak a nyílt szabványok használatára és a nyílt elérést biztosítják. Ezen kezdeményezéshez csatlakozott a Debreceni Egyetem Egyetemi és Nemzeti Könyvtár által fejlesztett elektronikus folyóirat-kezelő rendszer, a Métisz.

Februári működése óta 3 folyóirat csatlakozott a rendszerhez, és további számos felkérés érkezett. A kezdeményezést a Debreceni Egyetem Rektora, Prof. Dr. Fésüs László indította, a rendszerben elsőnek megjelenő tudományos folyóirat, a Gerundium egyetem-történeti közlemény volt.

A Métisz programozója és fejlesztőjeként lehetőségem adódott meghatározni a követendő elveket és szabványokat. A programozás során felmerülő kérdésekre gyors, határozott és a lehető legjobb választ kellett adni a viszonylag szűk határidő miatt.

Egyik fő célom a könnyű kezelhetőség, erről bővebben a (7) fejezetben olvashatunk. Programozás során figyeltem a program átláthatóságára, a függvények függőségének csökkentésére, a rendszer stabilitására és modularitására a program főbb függvényeiről a (6) fejezetben olvashatunk részletesebben.

Feltehetjük a kérdést: „Miért kellene ilyen rendszerek?”, a tudományos publikációk megjelenését biztosító folyóiratok egyre drágábbak, a technikai újítások vagy az elektronikus kiadások sem csökkentették az árakat. Az költségek csökkenése addig nem fog bekövetkezni, amíg az „információs piac” ezen szegmensét szinte csak és kizárólag a profitorientált kiadók birtokolják. Ezzel megteremtve azt a problémát, miszerint igazságtalan helyzet, hogy az egyetemek oktatói és kutatói által előállított publikációkat, amelyeknek kutatását gyakran közpénzből finanszírozzák, csak fizetős folyóiratokban olvashatjuk. Ezen kérdésre számos válasz és megoldás született és születik: egyik megoldást a Métiszhez hasonló rendszerek jelenthetik, de vannak kezdeményezések arra is, hogy egy bizonyos idő elteltével a kiadók a folyóiratokban megjelenő cikkeket ingyen közzéteszik egy nagyobb adatbázisban.

A Métisz tehát valójában nem csak egy rendszer, a Métisz egy irány, egy javaslat, amelynek helyességét majd a jövő dönti el és bírálja.

2 A három „O”: Open Source, Open Standards, Open Access

Ebben a fejezetben bemutatom, hogy mit takarnak ezek a kifejezések, hogyan alakulhat a jövőjük, és mi történt eddig.

Már az internet és a számítástechnika megjelenése előtt is voltak ilyen kezdeményezések. Példának okáért a könyvtáros világban a nyitott szabványnak, más néven Open Standardnak tekinthető, mikor 1877-ben az ALA (Amerikai Könyvtáros Egyesület) 7,5 cm X 12,5 cm-ben szabványosította a katalóguscédula méretét. Tehát e kifejezések egy része, már az internet vagy a számítástechnika megjelenése előtt megjelent, de a mai társadalmi, technikai, és politikai helyzetben rengeteg kérdés és probléma merült fel. (ALA, 1989)

2.1 Open Source – Nyílt rendszerek

Ez a kifejezés olyan szoftvereket takar, amelyek általában ingyenesek, és megkapjuk a szoftver forráskódját is, ezzel lehetőséget kapunk a szoftver saját fejlesztésére, csiszolására, javítására. A szoftver terjesztését nem akadályozza semmi, nem lehetnek diszkriminációk. Az általam fontosnak tartott előnyei:

- A szoftvereket ingyenesen megkaphatjuk, megszerezhetjük.
- Bármennyi gépre feltelepíthető egyszerre, mert a szoftver ingyenes.
- A szaktudás miatt a karbantartásukhoz és telepítésükhöz nem kell céggel szerződni.
- Az ingyenesség miatt bármeddig tesztelhetjük, nincs időkorlát.
- A nyílt forráskód miatt saját javításokat hozhatunk létre.
- Ha hibát találunk, akkor azt mi magunk is javíthatjuk.
- Minél többen dolgoznak rajta, annál több javítás és új alkalmazás jön létre.

A nyílt rendszerek természetesen több előnnyel rendelkeznek, de számomra az egyik legfontosabb, hogy egy cég vagy egyesület maximális erején felül, a használók egyéni megoldásai tovább bővítik a lehetőségek tárházát. Sajnos ebből adódóan, ha nincs egy központi „erő” ami irányítja a fő fejlesztést akkor a rendszer zavarossá, vagy akár használhatatlanná válhat. (Tibor, A nyílt rendszerek, 2007)

2.2 Open Standards – Nyitott szabványok

Pontos definíciót meghatározni nagyon nehéz, de nyitott szabványról beszélhetünk használati értelemben, ha bárki felhasználhatja a szabványokat szoftverfejlesztésre, és a felhasználók valamilyen módon részt vehetnek tovább fejlesztésükben. Ebből adódóan a szabványokat nem védi szabadalom, használatukhoz nem kell gyári szoftver. Ilyen típusú nyílt szabvány az **XHTML**, de ha részt akarunk venni alakításában, akkor csatlakoznunk kell a **W3C** konzorciumba, ami tagdíjfizetéssel lehetséges. A Dublin Core azonban minden lehetséges, manapság használt definíciónak megfelel, például, ha részt veszünk valamelyik fejlesztői értekezleten, akkor segíthetünk a fejlesztésében is.

Azon fájlformátumok, amelyek nyitott szabványon alapulnak és megfelelően megtervezettek, sokkal megbízhatóbbak, időt állóbbak és kevésbé avulnak el. Ha mégis elavulnak, akkor egy újabb szabványra való átállás sokkal könnyebb, a konverzió nyitott és ingyenes. Ha ezeket számba vesszük, akkor rájövünk, hogy szinte csak az ilyen, és ehhez hasonló szabványok lehetnek megoldások a dokumentumok hosszú megőrzésére. És mindez megoldhatja azt a problémát is, amely az egyes fájlformátumok hardver- vagy szoftverfüggőségét okozza.

Könyvtári kezdeményezések hada célozza meg a nyitott szabványok használatát. Ilyenek az OAI (Open Archives Initiative) és azon belül az OAI-PMH (Protocol for Metadata Harvesting) a DOI (Digital Object Identifier) és a Dublin Core Initiative.

Dspace, a nyílt forráskódjával, a nyitott szabványokra való támaszkodásával az egyik legjobb példa, ami a Massachusetts Institute of Technology (MIT) és a Hewlett-Packard (Hp) Labs közös fejlesztése. A küldetés kinyilvánítása szerint az MIT teljes közösségének digitális repozitóriuma, a tanszékek, intézetek helyett központilag kezeli a kutatási anyagokat és publikációkat.

A nagyobb hírű intézetekben több mint tízezer dolgozat keletkezik évente. A rekordokat Dublin Core segítségével írják le, csak három mező kitöltése kötelező, cím, nyelv, beadás dátuma, a többi opcionális. (Tibor, 2007) A DSpace mint adatszolgáltató, interfészek sokaságával rendelkezik: SOAP, OAI, WSDL, JSON, így természetesen az előbb tárgyalt OAI-t is támogatja.

A LOCKSS a Stanford Egyetem projektje, amelynek a hosszú távú megőrzés támogatása a célja, egy intézmény által előfizetett elektronikus folyóiratok követését, frissítését, és archiválását látja el. (Tibor, 2007)

Az NDA (Nemzeti Digitális Adattár) 2003-ban indult, ez az adattár az OAI-PMH-t használja. A nyitott szabványok az XML és a metaadatok együttes felhasználásának hazai példája, amely a különböző szerverekről származó gyűjtemények közös lekérdezését szolgálja. A nyílt rendszerek és a szabványos metaadatok felhasználásával a közös lekérdezések jól megoldhatók.

Az **OAI** (Open Archives Initiative) társadalmi kérdésként kezeli az akadémiákban létrejövő információkat, mivel azok közkincsnek számítanak, így egyértelmű a nyílt hozzáférés. Egy egységes keretrendszert tud ajánlani a csatlakozó intézményeknek, így gyűjteményeik könnyen felkereshetőek.

A keresés központjában az OAI-PMH áll. Az adatok a szolgáltatók, intézmények gépein vannak, sőt azok metaadatai is. A központi szerver átadja a kérdést, majd a visszakapott metaadatokat a kérdező felé szolgáltatja. A metaadatokban általában megtalálható egy közvetlen link a dokumentum eléréséhez. Ez a rendszer így teszi lehetővé, hogy a kisebb intézményeknek, könyvtáraknak ne kelljen külön megjelenítő, kereső felületet fejleszteni, karbantartani és üzemeltetni, csak a gyűjtésre, és a pontos leírásra kell koncentrálniuk.

Az NDA OAI szerverek (JAVA, PHP stb.) és dokumentációk letöltését kínálja. Tagja és állományai kereshetők az Országos Széchényi Könyvtárban, a Fővárosi Szabó Ervin Könyvtárban, a MEK-ben, MATARKA-ban, a Neumann János Digitális Könyvtárban, de idetartoznak olyan oldalak is, mint az ERIKANET (Egységes Regionális Információs Közművelődési Adatbázis), a Keresztény múzeum, a Duna Múzeum vagy a Litera.hu vagy több filmes portál. (Tibor, A nyitott szabványok, 2007)

2.3 Open Access – Nyílt hozzáférés

A tudományos publikáció, a tudományos folyóirat-kiadás megváltozására irányuló kísérleteket összefoglaló néven nyílt hozzáférésnek nevezzük.

Az elektronikus folyóiratok által felmerült kérdések egyik fontos vagy sokat vitatott pontja, a nyílt hozzáférés. Az elmúlt évek tendenciái azt mutatják, hogy mind a nyomtatott, mind az

elektronikus folyóiratok ára növekszik. Ez több tényezőnek is betudható, de az egyik és legmeghatározóbb, hogy a folyóiratok technikai és megjelenítési módjai már rég megváltoztak, az erőviszonyok átrendeződtek. De ezt nem követte a gazdasági keret megváltozása, így sajnos az információs piacon egyeduralommal rendelkező kiadókat a technológiai fejlődés még erősebbé és nagyobbá tette. Bár az emberek többsége azt feltételezi, hogy az elektronikus folyóiratok olcsóbbak, ez a feltételezés hamis, mivel a kiadók a maximális profitra törekszenek és a profitorientáció, illetve az egyeduralom lehetővé teszi számukra az árak szinten tartása helyett, a kiadványaik árának folyamatos növelését. Sajnos a kiadók azt a határt próbálják megtalálni, amit még az olvasó hajlandó és ki is tud fizetni, ezen okokból - ahogy azt már kifejtettem az előbbiekben - az elektronikus folyóiratok ugyanolyan drágák, mint a nyomtatott változatok, így a folyóiratválság még ma is él, sőt egyre nagyobb probléma a könyvtárak mindennapi életében.

Ezen okokból megjelent az etikai dilemma is, ha belegondolunk az egyetemen létrejövő tudományos kutatások eredményeit, az adófizetők pénzéből finanszírozták/finanszírozzák, ezek publikációját viszont magáncégek kezelik és irányítják. A kutatások eredményének megismeréséért egy másik kutatónak – akinek a kutatási költségeit az állam fizeteti - ismételtén fizetnie kell. Ennek megszűnése érdekében több alternatív publikációs modell jött létre.

Az internet rengeteg új és eddig ismeretlen lehetőséget nyújt - amellyel sokan éltek - a próbálkozások a drága kiadók kikerülését, kihagyását célozták meg. Az 1990-es években számos elektronikus preprint-archívum és néhány száz lektorált elektronikus folyóirat jött létre. A cikkek ingyenes elérése mellett a legfontosabb az ilyen folyóiratok és archívumok esetében, hogy a cikk teljes tartalma elérhető. Ezek a folyóiratok általában egyéni kezdeményezésre, kizárólag elektronikus formában jelentek meg, mert így lehet a költségeket legjobban minimalizálni.

Hernád István (Stevan Harnad) mutatott rá: „A folyóiratokban közölt cikkek zöme egy szűk szakmai körnek szól.” és a papíralapú publikálás kényszerűsége, hogy az ilyen típusú cikkeket, publikálásokat azonos módon kezeljük a jövedelemszerzésre irányuló sajtótermékekkel, de az elektronikus folyóiratok lehetőséget adnak ennek a problémának a kiküszöbölésére is.

Évi 2,5 millió cikkről beszélhetünk, amelyeket 24 ezer lektorált folyóiratban a legkülönbözőbb tudományterületeken, nyelveken és országokban publikálnak. (Harnad, 2005)

Mi pontosan akkor a nyílt hozzáférés? Definícióját a következő pontokkal lehetne összefoglalni:

- A felhasználónak nem kell fizetnie azért, hogy tudományos közleményeket közvetlenül olvashasson, azokat korlátlanul elérhesse, nem kereskedelmi célból kinyomtathassa.
- Harmadik fél számára tilos a tartalom felhasználása.

Nyílt hozzáférés esetén a szerző a szerzői jogokat magának tartja meg és máshol is közzé teheti műveit.

Ezek megvalósítási formája sokféle lehet. Az általam fontosabbak a következők:

1. Korlátozás nélküli hozzáférés (folyóiratok azonnali és teljes publikálása)
2. Késleltetett nyílt hozzáférés (a nyílt hozzáférés csak egy idő után valósul meg.)
3. Szerzői díjas hozzáférés (a szerzők díjat fizetnek a nyílt hozzáférés támogatása miatt.)
4. Részleges hozzáférés (Csak a cikkek egy részéhez biztosítanak hozzáférést)
5. Kooperációs hozzáférés (Intézményi tagok támogatják a nyílt hozzáférést.)

(Corrado, 2005)

A lektorált tudományos-elektronikus nyílt folyóiratok fontos részei továbbá:

- Aktuális területenként szerveződő preprint-archívumok (tematikus repozitóriumok)
- Az egyes egyetemek intézményi repozitóriumai.
- A publikációknak a szerzők saját honlapjain történő elhelyezése és archiválása.

(Björk, 2004)

A **preprint** olyan tudományos publikáció, amely még nem esett át lektoráláson.

A **posztprint** ezzel szemben lektorált és publikált, amelyeket általában a kiadók engedélyével tesznek közzé a kutatók.

Az elektronikus preprinteket és posztprinteket együttesen **E-printeknek** nevezzük. Elhelyezésüket pedig nyílt archívumban *ön-archívumnak* nevezzük. (Zsolt, 2003)

Preprintek olyan kutatási területeken vannak jelen, ahol már ennek korábbi, a számítástechnika megjelenése előtt is voltak hagyományai, ennek oka az új információ áramlásának sebessége. Az ilyen archívumok működési alapelve, hogy a kutatók maguk teszik fel a repozitóriumokba a kézírataikat, ezzel humánerőforrást szabadítanak fel, ami így a költségek csökkentéséhez vezet, ezáltal elérhető az igen alacsony fenntartási költség.

A legújabb nyílt hozzáférésű csatornák, az intézményi repozitóriumok. Az egyetemek és a könyvtárak informatikai háttere sokkal megbízhatóbb és stabilabb, mint egy kutató vagy egy kis intézmény háttere, ezáltal nagyobb biztonsággal képesek a gyűjtemények hosszú távú megőrzésére és szolgáltatására.

Az egyik fontos állomása a nyílt hozzáférésnek Paul Ginsparg nevéhez köthető, aki 1991-ben létrehozta és elindította az arXiv preprint szervert. Három esztendő múlva egy kedélyeket borzoló és az addigi szemszögtől teljesen eltérő javaslat, álláspont látott napvilágot: az internet korában nincs szükség arra, hogy a kutatók cikkeit az egyre drágább kiadók adják ki, ezért a kutatók maguk hozzák nyilvánosságra a publikációjukat.

Kiemelném a nyílt hozzáférésű archívumok közül a PubMed Central-t, 2000-ben jött létre, amely egy ingyenes online teljes-szöveges archívum, ahova a kiadók önkéntesen töltenek be adatokat, a megjelenést követő néhány hónap után - ezt a megoldást nevezzük késleltetett nyílt hozzáférésnek.

Egy 2004-es nemzetközi felmérés tízezer kutatót vizsgált, ennek háromnegyede tudott valamit a nyílt hozzáférésről, 14%-a úgy ítélte meg, hogy sokat, míg 48%-a keveset tud róla. A válaszadók 46%-a még sohasem közölt cikket nyílt hozzáférésű folyóiratban. (Nicholas, 2005.)

A nyílt hozzáférés előtt álló számos akadály közül csak kettőt emelünk ki:

- a megfelelő üzleti modellek hiánya
- az akadémiai előmeneteli rendszer. (Björk, 2004)

A nyílt hozzáférésű elektronikus folyóiratok törekednek a befektetett energia minimalizálására, ennek több oka van: a nyílt folyóiratok sajnos a legtöbb esetben egyéni vagy kiscsoportos kezdeményezések, így anyagi és humán erőforrásai nagyban kötöttek. Ha a résztvevőket a munkáltatók fizetik, tehát ezt a munkakörükön belül végzik, akkor a

mindennapi munkától vonnak el energiát. Nem nehéz tehát továbbgondolni, hogy ez a struktúra csak abban az esetben jó, ha az adott folyóirat éves cikkszámja alacsony. (Tibor, A nyílt hozzáférés, 2007)

Vannak természetesen más üzemeltetési struktúrák, de annyira széles a lehetőségek tárháza, hogy bemutatása talán egy másik dolgozat témájának is megfelelne.

Miért csak ezt az egy struktúrát emeltem ki? Mert mi is hasonló üzleti modellben üzemeltetjük a Métiszt – az általam fejlesztett elektronikus folyóiratkezelő-rendszert – egy kis módosítással.

3 Métisz?

Miért pont Métisz lett a projekt neve? **Métisz** (görögül: Μήτις) az *értelem és a megfontoltság* istennője (Métisz, 2010). Nem is találhattunk volna szebb nevet ennek a projektnek, hiszen a cél az, hogy a Debreceni Egyetemen lehetőség legyen nyílt folyóiratok – *értelem* – megjelentetésére. A rendszer fejlesztésénél nagyon fontos volt a megbízhatóság, a nemzetközi tendencia átgondolt követése – *megfontoltság* – és az energiatakarékosság, ez utóbbiban még fejlesztésekre van szükség, de már most is sokkal könnyebb ezt a rendszert kezelni, mint a hozzá hasonló rendszereket. (7.1)

3.1 Miért saját?

A projekt már elkezdődött, amikor én csatlakoztam a fejlesztéséhez, így az előzményeket a munkatársaim feljegyzéseiből ismertem meg. A kérdés az, hogy „Miért saját?”. Már a kezdetektől felmerült és rengeteg érv és ellenérv sorakozott fel egymással szemben, a legfontosabbak:

Kérdés: Miért nem jók a már létező nyílt rendszerek?

Válasz: A létező rendszerek közül soknak a feltelepítése is kész művészet volt, és egyeseket fel sem tudtunk telepíteni, míg más rendszerek nem tudták teljesíteni az elvárásokat, amiket felállítottunk, vagy a kezelése olyan bonyolult volt, hogy a felhasználók nehezen birkóztak volna meg vele. Végül az OJS (<http://pkp.sfu.ca/?q=ojs>) nyert. A rendszer a legfontosabb kritériumokat teljesítette, de feltelepítés után a tesztidőszakban hamar előjöttek azok a problémák, amik a tökéletes működés érdekében szükségesek, ezeket a hibákat részletezem egy későbbi fejezetben.

Kérdés: Másoknak jó, nekiünk miért nem?

Válasz: Az ilyen típusú kérdésekre mindig azt válaszolom magamban, hogy mert mi nem mások vagyunk, nem mindegy, hogy egy rendszert milyen környezetre készítenek. A környezetbe beleértendő például a használók informatikai tudása, az általános munkamenet, az ott használt minimális adatok, azok mennyisége, részletessége, stb. Egyszerűen a magyar viszonyok ilyen tekintetben nem egyeznek meg a külföldiekkel, így a rendszer nem képes maradéktalanul ellátni a feladatát. A következőkben majd példákkal is igazolom a fent említetteket.

Kérdés: Stabilabbak a nyílt rendszerek?

Válasz: Szerintem a nyílt rendszerek változó stabilitással rendelkeznek és nem csak rendszer és rendszer között értem, hanem egyes verziók között is lehetnek ilyen stabilitási problémák, így nem lehet olyan kijelentést tenni, ami azt mondja, hogy minden nyílt rendszer stabil, vagy stabilabb, mint amit projektben a könyvtár maga fejleszt.

Kérdés: Könnyebb a fejlesztése?

Válasz: A válasz nem egyértelmű, igaz vannak rendszerek, amelyekhez könnyebb saját modult vagy javítást készíteni, és ha elküldjük, akkor elképzelhető, hogy berakják a csomagba. Lehetséges az az eset is, amikor már létezik a probléma megoldása, csupán utána kell olvasni, és letölteni a javításokat vagy a modult.

Kérdés: Ezen tulajdonságokkal a rendszer is rendelkezik?

Válasz: Szerintem egy rendszer vagy honlap akkor jó, ha tökéletesen biztosítja azt a feladatot, amire tervezték, de ez az elsődleges funkció a legfontosabb, és csak akkor lehet a rendszert egyéb lehetőségekkel felruházni, ha ezt a funkcióját nem veszélyezteti. Nagyon fontos az is, hogy a másodlagos funkciókat is tökéletesen el tudja végezni. Ha ez nem történik meg akkor szinte biztosan állíthatom, hogy hibás döntés az, ha ezt erőltetjük, főleg azzal az indokkal, hogy így csak egy felületet kell használni.

3.2 A Métisz elődje (OJS hibák)

Az előző részben csak megemlítettem, ebben a fejezetben részletesebben kifejtem az OJS rendszer tulajdonságait és hibáit. Az OJS (<http://pkp.sfu.ca/?q=ojs>) azért jött létre, hogy erősítse a nyílt hozzáférésű elektronikus folyóiratok létrehozását és működését. Az OJS egy biztos rendszer abban az esetben, ha illeszkedik a terveinkhez és elvárásainkhoz, de ha módosítása szükséges valamilyen probléma megoldására, akkor falakba ütközünk. A rendszer PHP nyelven íródott és használja a sablonokat is, amelyek a rendszer gyorsítását szolgálják. Ezáltal sajnos maga a kódszerkezet is sokkal bonyolultabb lesz és kisebb lekérdezés szám esetén nem is kifizetődő. A pontos okot ennek használatára talán csak a fejlesztők tudnák megmondani.

Azt tapasztaltuk, hogy a rendszer igaz, képes egymás mellett lévő folyóiratok üzemeltetésére, de sajnos nem képes a folyóiratok adatainak teljes körű elkülönítésére; a rendszer automatikus leveleket küld, felkérésnél és minden fontosabb eseménynél, de sajnos ezeknek a leveleknek a szövege minden folyóiratnál megegyezik. Bizonyos adatok: folyóiratnév, szerző neve, cikk neve természetesen automatikusan változik, ez hatalmas hiba, hiszen ha belegondolunk, minden tudományterületnek saját nyelvezete van.

További nehézséget jelentett, hogy az űrlapokat semmilyen módon nem lehetett szerkeszteni, csupán ha a forráskódot is módosítottam, természetesen ez akkor hatásos megoldásnak bizonyult, de csak abban az esetben, ha alapértelmezett értékekkel töltjük ki az eltüntetett mezőket. Tehát azt viszonylag könnyen meg tudtam oldani, hogy a nem kívánt űrlapmezőket kivegyem, vagyis jobban mondva eltüntessem, de újabb mezők bevezetése sokkal nehezebb. Nem lehetetlen, de nagyon nehézkes és akár az egész rendszer integritását megváltoztatta volna. Például az adatbázis struktúrája nem volt arra alkalmas, hogy a cikkek nevét magyarul és angolul is eltárolhassuk. Ezt a problémát is megoldottam, de az a fő függvények átírásával járt, mivel a cikkek neve ott generálódott és alakultak át linkké.

A beküldendő fájlknál sajnos nem tudta a program alapértelmezetten megvalósítani azt, hogy csak bizonyos fájltypusokat engedélyezzen feltölteni. Ezt mondanom sem kell, hogy mekkora hiányosság, hiszen vannak olyan területek, mint például a TDK, ahol csak pdf fájlokat lehet elfogadni a dolgozat utólagos módosítása és védelme érdekében.

Másik nagy hiba volt az, hogy egyes munkafolyamatok túlzottan részletesek voltak, mondhatnám apró részletfolyamatokig. Más munkafolyamatok viszont pont ellenkezőleg, elég szegényes mozgáskörrel rendelkeztek. Nagyon zavaró volt, hogy egy cikknek mindenképpen minden munkafolyamaton át kellett esnie, de ha nem tudunk alkalmazni lektort, vagy korrektort vagy nincs szükség tördelésre, a cikknek akkor is át kell esnie a procedúrán, ami elsőre nem is hangzik soknak, de ha belegondolunk, hogy el kell küldeni a szerkesztőnek minden munkafázisra az adott cikket, és ott ismét végig kell kattintgatni, ez több cikk esetében idegörlő.

Nem tartom hibának, de számomra hátrányt jelentett a rendszer bonyolult menüszerkezete. Ha számomra - gyakorlott informatikai felhasználó - nem volt egyértelmű, és még megmutatás után is problémát okozott egy cikk feltöltése vagy bírálatra küldése,

gondoljunk bele, hogy egy olyan embernek, akinek nem az informatika és az informatikai rendszerek kezelése, használata a fő feladata, mekkora kihívást jelentene. Ha nem vagyunk képesek olyan egyszerű és könnyen kezelhető rendszereket létrehozni, amelyek segítségével játszani könnyedséggel lehet feltölteni és figyelni a cikket, akkor ne is várjuk el azt, hogy a szerzők a már megszokott rendszertől megválnak és átállnak a mi nyílt elektronikus folyóiratainkra.

Összefoglalva, az OJS egy nagyon jó rendszer, kis hibáit leszámítva, amik viszont szinte mindegyik vetélytársában megvannak, és ha tudásánál nem akarunk semmivel sem többet, mint amit tud, akkor szerintem mindenképpen érdemes a kis listánkra felírni, amik közül a végső győztest fogjuk kiválasztani.

4 A php és egyéb szolgáltatások

Ebben a fejezetben a rendszerhez csatolt szolgáltatásokat részletezem, amelyek segítik a rendszer beépülését a könyvtár informatikai arculatába. Először bemutatom a PHP nyelvet, amin a rendszert írtam, majd a hozzá közel álló egyéb rendszereket, szabványokat.

4.1 PHP

A Métisz PHP programnyelvben készült. A PHP működése leegyszerűsítve

„A PHP oldalak elkészítésénél a HTML-t gyakorlatilag csak mint formázást használják, ugyanis ezen lapok teljes funkcionalitása a PHP-re épül. Amikor egy PHP-ben megírt oldalt akarunk elérni, a kiszolgáló először feldolgozza a PHP utasításokat és csak a kész HTML kimenetet küldi el a böngészőnek, így a programkód nem is látható kliens oldalról. Ehhez egy úgynevezett interpretert (értelmezőt) használ, amely általában egy külső modulja a web-szervernek.

A PHP nyelv lényegében nagymértékű kiegészítése a HTML-nek, ugyanis rengeteg olyan feladat végezhető el vele, amelyre az ügyféloldali szkriptek nem képesek, vagy ha igen, akkor csak korlátozottan. Ilyen például a bejelentkezés, az adatbázis kezelés, fájlkezelés, kódolás, adategyeztetés, kapcsolatok létrehozása, e-mail küldése, adatfeldolgozás, dinamikus listakészítés stb. Minden olyan esetben, ahol nagyszámú ismétlődő feladatsort kell

végrehajtani - például képek listázása és linkelése, listakészítés stb. - ott ez a programnyelv nagyszerű segítség.” (PHP)

4.2 LDAP

„Az LDAP a Lightweight Directory Access Protocol rövidítése, mint a neve mutatja, az LDAP nem több egy protokollnál, nem egy konkrét szoftverről van tehát szó. Ez a protokoll directory szolgáltatások elérését szabályozza.,, (Zoltán, 2003)

„Az angol *directory service* egy olyan speciális adatbázist takar, amely keresésre van optimalizálva, ennek megfelelően olyan esetekben célszerű használni, ahol kevés a módosítás és nagyszámú, gyors lekérdezésekre van szükség.” (Zoltán, 2003)

Az egyetem és a könyvtár a felhasználók nevét, jelszavát és némi egyéb adatot az LDAP-ban tárol, így lehetséges az, hogy különböző a szolgáltatásokra ugyanazzal a felhasználó névvel és jelszóval jelentkezünk be, de ennek a rendszernek további számtalan előnye van.

4.3 SOAP

„A számítástechnikában a **SOAP** egy üzenetküldésre használt, XML-alapú formátum, a SOAP formátumot használó üzenetek azután különböző protokollok segítségével továbbíthatók, de erre a célra általában a HTTP protokollt használják.

A SOAP, a WSDL és az UDDI mellett a web szolgáltatások harmadik alapvető elemét képezik. A név eredetileg az angol *Simple Object Access Protocol* rövidítése volt, de a szabvány 1.2-es verziójától kezdve a *Simple Object Access Protocol* nevet már nem használják, mert az elnevezés félrevezető volt. A SOAP szabványt a W3C tartja karban.” (SOAP)

4.4 A DEA

A Debreceni Egyetem Elektronikus Archívum (DEA), természetesen a Métisz SOAP (4.3) segítségével a cikkeket, folyóiratszámokat és az egyéb adatokat a DEA-ba archiválja, ezzel biztonságba helyezi az itt keletkező anyagokat, és egyúttal erősíti a DEA-t. Nem beszélve arról, hogy a DEA-n keresztül is elérhetőek a cikkek (ha a beállított jogosultságok

engedélyezik). A legfontosabb, hogy az adatok szabványosan vannak tárolva, így a rendszer kihasználja a nyílt szabványok (2.2) előnyeit, és biztosítja az adatok rendszerfüggetlenségét.

5 A Métisz maga

Ebben a fejezetben a forráskód egyes részleteit szeretném taglalni, és a rendszer fontosabb csomópontjait, a fejlesztés alapelveit, a rendszerhez csatolt szolgáltatásokat és azok hasznosságát.

Az egyszerűség miatt innentől a Métisz néven nem csupán a projekt nevét értem, hanem az egész forráskódot is. A forráskód megismertetése előtt mindenképpen a főbb gondolatok és irányelveket kell áttekintenünk.

5.1 Irányelvek, gondolatok, megvalósítások.

Ebben a fejezetrészben csak az alapelveket és azok körvonalait fogalmazom meg, a megoldást, és a megvalósítást a hasonló nevű fejezetben olvashatjuk. A rendszer legfontosabb alapelveit részletezem:

- könnyű laza adatszerkezet
- könnyű, és egyszerű kezelés
- beállítások széles tárházát biztosítja
- munkafolyamatok jó elkülönítése
- munkafolyamatok szerkesztő általi irányítása
- rendszerfüggőségek csökkentése
- jogosultságok könnyű kezelése
- jogosultságok folyóíratonkénti lehetőségeinek szabályozása
- többnyelvűség megteremtése
- generált szövegek folyóíratonkénti elkülönítése
- gyorsaság és stabilitás
- rendszer-folyamat rugalmassága.

5.2 Könnyű laza adatszerkezet

Az adatbázist olyan módon kellett megtervezni és létrehozni, hogy bármilyen plusz adatot hozzárendelhessünk a cikkhez, vagy adott esetben a szerzőhöz. Ez a pont mindenképpen a rugalmasságot szolgálja, hiszen minden folyóirat maga dönti el, hogy milyen adatokat szeretne bekérni a szerzőktől és mindezt a többi folyóirattól függetlenül.

A következő megoldást alkalmaztam: a főtablák mellé létrehoztam „puffer” táblákat, amelyben a részletes adatokat tároltam; egy adatról tárolom, hogy melyik cikkhez tartozik és az adat pontosan milyen jellemzőkkel írható le: név, cím, szám, stb. Ez lehetőséget ad bármilyen adat bekérésére, hiszen ez az adatbázis struktúráját nem borítja fel.

5.3 Könnyű és egyszerű kezelés

A rendszer kezelhetőségét nem csak a felhasználói szerepből kell figyelni, sokan ezt az igen fontos szempontot elfelejtik. A karbantartást nagyban megnehezítheti egy bonyolult szerkezet: például a rendszer fordítása nagyon egyszerű és egyszerre akár többen is fordíthatják a rendszert. Ennek pontos leírását képi illusztrációval a későbbiekben fejtem ki. Több tesztet is végeztünk, amelynek eredményeképpen megtaláltuk a legoptimálisabb elrendezést: az űrlapok adatbekérő mezőit nagyobb csoportokba szedtük össze, ezzel növelve az átláthatóságot és értelmezhetőséget.

5.4 A beállítások széles tárházát biztosítani

A rendszer adminisztrációs felületének lényege, hogy a jogosultságokkal rendelkezők a lehető legtöbb funkciót saját maguk, programozó beavatkozása nélkül is beállíthassák. A különböző folyóiratok egymástól jól elkülöníthetők és szinte minden átszabható folyóiratonként. A teljes kinézet és menüszerkezet átszabásán felül, a jogosultsági szinteket is folyóiratonként megszabhatjuk. Ez pontosan mit is takar? További fontos beállítás az automatikusan generált levelek gyors és egyszerű megírása, módosítása. A menük elhelyezése, és szövege mellett azok tartalmát is könnyen egy-két kattintással szerkeszthetjük: impresszum, vagy általános tájékoztató, sűrű.

5.5 A munkafolyamatok elkülönítése

A munkafolyamatok könnyen elkülönülnek: a menüben egymás alatt felsorolva találhatók a munkafolyamatok, itt mindig csak a jogosultságnak megfelelő linkek látszanak, így egy szerzőnek csak 3 link látható: kilépés, profil és szerző - egy bírálónak kibővül a Bíráló linkkel. A munkafolyamatok függetlenek egymástól, jobban mondva egyik munkafolyamat sem indokolja az előző meglétét. Ha viszont szeretnénk egy cikknek külön HTML oldalt, ehhez a cikk nem olyan állapotban, formátumban van, ami a folyamatot azonnal lehetővé tenné, akkor a tördelői munkafázis kihagyhatatlan.

5.6 Munkafolyamatok szerkesztő általi irányítása

A munkafolyamatok száma és egymásutánisága nincs meghatározva, a rendszer alapesetben nem ellenőrzi, ezt a szerkesztő vagy szerkesztők ellenőrzik, ezzel elérve, hogy egyszerre több bírálónak is elküldhető a cikk, vagy éppen teljesen kihagyható. A cikk adatai közé automatikusan beíródnak a cikkel történt események (mikor? ki? mit?). A szerkesztő bármikor eldöntheti, hogy a cikk melyik számban, annak melyik rovatában jelenjen meg, vagy akár megjelentetheti a szám kiadása előtt is a cikket, lezárhatja a számot, így már további cikkek nem adhatóak hozzá, de ekkor még a kiadás nem történik meg; további műveletek a cikkekkkel viszont lehetségesek.

5.7 Rendszerfüggőségek csökkentése

Törekedtem arra, hogy a rendszer működését meghatározó külső elemek mennyisége minél kisebb legyen. A rendszernek alapvetően a PHP futtatási környezetre és egy MySQL adatbázisra van szüksége. Ezen felül természetesen további modulok, alkalmazások csatolhatóak hozzá. A rendszerfüggőség nagy gond lehet, ha egy rendszer alapvető működése több rendszertől függ, veszélyt jelent a működés szemszögéből és biztonsági kockázatot is jelenthet a rendszer számára.

5.8 A jogosultságok könnyű kezelése

A legtöbb rendszerben jogosultsági csoportok vannak: felhasználó, admin, segítő, root, stb, ezen csoportoknak meghatározott lehetőségei vannak. Természetesen ez a Métiszben is így

van, azzal a különbséggel, hogy itt egyes felhasználóknak, vagy csoportoknak átdefiniálhatjuk a jogait, ami részletekig lehetséges. Így a jogosultságokon keresztül korlátozhatjuk, hogy a rendszer kiknek mutassa a bírálók nevét, vagy a cikkek adatsortjait, stb. Könnyen hozhatunk létre új szűkítéseket, ehhez igaz programozói segítség is kell, de nagyon könnyű leprogramozni. A jogosultsági függvények részletes leírása a dolgozatban később bemutatásra kerül.

5.9 A többnyelvűség megteremtése

A Métiszben nem csupán arról van szó, hogy egyik nyelvről a másik nyelvre lefordítjuk az egész oldalt, de a rendszer képes „nyelvjárásokra” is. Ez azért hasznos, mert minden folyóirat saját nyelvezettel rendelkezhet, így az adott szakterület terminológiáját felhasználhatja, ezáltal az eszközzel is a felhasználókat közelebb hozhatjuk a rendszerhez és a magyarázatok mennyisége és minősége is folyóiratonként változhat.

5.10 Gyorsaság, és stabilitás

A Métiszhez hasonló rendszereket „hivatkozási csomópontnak” nevezzük, mivel az információs forradalom miatt a publikációk egyre nagyobb százaléka elektronikus formában kerül publikálásra, a hivatkozások web helyekre és publikációs rendszerekre hivatkoznak. Ebből könnyen adódik az a helyes gondolat, hogy ezeknek a rendszereknek stabilnak és visszamenőleg a lehető legkompatibilisebbnek kell lennie. Abban is biztosak lehetünk, hogy az a rendszer, amelyik folyamatos leállással és problémákkal küszködik, sosem lesz közkedvelt, és még kevésbé szeretett. Sokan elfelejtik, hogy a mai kor egyik alapvető követelménye az egyszerűség mellett a gyorsaság. Senki sem szeret egy lekérdezés vagy valamilyen parancs kiadása után percekig ülni a gép előtt. Természetesen most sarkosítottam a problémát, de a lassúság igenis fontos szempont a felhasználók szemében. Tehát egy rendszernek ezt a két szempontot nem csak a fejlesztés során, de az egész működése alatt figyelembe kell vennie.

5.11 Rendszer folyamat rugalmassága

Mit is takarhat ez az érdekes kifejezés? A rendszert programozása során, az osztályokat, függvényeket, főbb eljárásokat úgy kell megtervezni, hogy ha szükséges ezek átírhatók legyenek, de azok módosítása ne követelje meg a rendszer teljes átírását. Egy példán keresztül

sokkal érthetőbb: a rendszerben használok egy függvényt, amely a lekérdezésben segít. A függvénynek megadhatjuk kérdésünket és visszaadja az adatbázistól kapott adatokat. Ez azért hasznos, mert ha más adatbázis szerverét, vagy más adatbázist használnánk, akkor nem kell a rendszerben tömeges módosításra és átszerkesztésre felkészülnünk, csupán a fő függvényt átírjuk, és a rendszer mehet is tovább.

Fontos, hogy a függvények lebontottsága legyen annyira egyszerű és követhető, hogy bárki könnyen módosíthassa a rendszert. Ez csak akkor lehetséges, ha kezébe adjuk a „lego kockákat” amik összeillesztésével egyszerűen létrehozhatunk új részeket vagy módosíthatjuk a meglévőket.

6 Függvények, csomópontok

Egy jó rendszer működéséhez hasznos építőelemekre van szükség, ha ezek az elemek átláthatóak és részletgazdagok, akkor a fejlesztés vagy javítás bárki számára nagyon könnyű. Nézzük tehát a rendszer alapvető építőköveit és annak függőségeit.

6.1 Adatbázis függvények

6.1.1 lekerdezes()

A lekerdezes(), egy függvény, amely az adatbázis-lekérdezés eredményét adja vissza tömb formájában.

6.1.2 létrehozás()

A létrehozás(), egy függvény, amely a már meglévő táblákba képes sorokat beszúrni. Ellenőrzi, hogy a létrehozásra van-e joga a felhasználónak az adott táblában, ha nem, vagy ha hiba következik be, akkor a hibag() függvény lefut a megfelelő paraméterekkel, ha sikeres a beillesztés, akkor pedig logol() függvény fut le.

Függőség:

- jogom()
- logol()
- hibag()

6.1.3 modositás()

A modositás(), egy függvény, amely a létező adatok módosítására képes. Ellenőrzi, hogy a módosításra van-e joga a felhasználónak az adott táblában, ha nem, vagy ha hiba következik be, akkor a hibag() függvény lefut a megfelelő paraméterekkel. Ha sikeres a módosítás, akkor pedig logol() függvény fut le.

Függőség:

- jogom()
- logol()
- hibag()

6.1.4 torles()

A `torles()`, egy függvény, amely a létező adatok törlésére szolgál. Ellenőrzi, hogy a törlésre van-e joga a felhasználónak az adott táblában, ha nem, vagy ha hiba következik be, akkor a `hibag()` függvény lefut a megfelelő paraméterekkel. Ha sikeres a törlés, akkor pedig `logol()` függvény fut le.

Függőség:

- `jogom()`
- `logol()`
- `hibag()`

6.2 Jogosultsági függvények

6.2.1 jogom()

A `jogom($kerdes)`, egy olyan függvény, amely segítségével lekérdezhetjük azt, hogy az adott joggal rendelkezik-e az aktuális felhasználó vagy sem. A függvény visszatérési értéke igaz/hamis.

Függőség:

- `lekerdezes()`

6.2.2 cikjoga()

A `cikjoga($sz_adatok, $cikk_id="0", $nalakelllenie="nem")` olyan függvény, amely ellenőrzi, hogy a megadott felhasználó-cikk páros létezik-e vagy sem. Megmutatja, hogy a felhasználónál van-e vagy volt-e a cikk. A harmadik paraméterrel azt állíthatjuk be, hogy a cikknek jelenleg az adott felhasználónál kell lennie vagy sem. A függvény visszatérési értéke igaz/hamis.

Függőség:

- lekerdezes ()

6.3 Megjelenítési függvények

6.3.1 linkg()

A `linkg($linkneve,$url_kiegeszites, $kiegeszites)` olyan függvény, amely segítségével generálhatjuk a linkeket. Ez a függvény már összetett, mivel ellenőrzi a jogokat, és a link nevének kiírását is a `kiirg()` függvényen keresztül végzi. Három bemeneti paramétere van: az első a link neve, ami egy azonosító kulcs általában, a másodikban azt adhatjuk meg, hogy milyen URL kiegészítést kapjon a link, a harmadik paraméter pedig a `<a>` tagnak adhatunk meg attribútumokat. Visszatérési értéke a generált link: „szövegesen”.

Függőség:

- jogom()
- kiirg()

6.3.2 kiirg()

A `kiirg($kulcs)` egy olyan függvény, amely a kulcsokhoz rendelt szót vagy kifejezést adja vissza „szöveg” formátumba. Ha nem létezik a kulcs, akkor a megadott kulcs jelenik meg.

Függőség:

- lekerdezes ()

6.3.3 hibag()

A `hibag($kulcs)` függvény a kulcsban megadott hibát írja ki, és meghívja a `logol()` függvényt, amely elmenti a hiba helyét, idejét és további adatokat. Visszatérési értéke a hiba megfogalmazása a `kiir()` függvényen keresztül, „szöveg” formátumban.

6.4 Log függvények

6.4.1 logol()

A logol(\$esemeny, \$ideiglenes_id="0"), függvény a rendszerben végbemenő cselekményekről és hibákról készít bejegyzést, az adatbázisba.

Függőség:

- lekerdezes ()
- létrehozás ()

6.4.2 clogol()

A clogol(\$esemeny, \$ideiglenes_id="0") függvény a cikkekről történt műveletekről készít log-t, így minden cikkel történt művelet visszakövethető.

Függőség:

- lekerdezes ()
- létrehozás ()

6.5 Fájlkezelő függvények

6.5.1 csatolmany_fel()

csatolmany_fel(\$mappa_kepnek,\$kep_felnev,\$kepadatok,\$linkelni,gyokermappa="atmeneti") függvény intézi a feltöltést, a függvény meghívásakor „szöveges” formában a feltöltő űrlap tér vissza és a tájékoztatás a feltöltésről. Az első paraméterben a mappa szerkezetet lehet megadni, amely a mappákat létrehozza. A második paraméter a változó nevét adja meg, a harmadik a \$_FILE['valtozo_neve'], itt válik láthatóvá a feltöltött fájl és annak adatai. A \$linkelni azaz a negyedik paraméterben azokat a fájlformátumokat lehet megadni, amelyeknek engedélyezzük a feltöltést. A fájlformátumokat MIME szabványban kell megadni és vesszővel elválasztani. A gyökérmappa alapértelmezetten meg van adva a függvény deklarálásánál, de átdefiniálható, ha szükséges.

6.5.2 csatolmányok()

A csatolmányok(\$mappa_kepnek, \$gyokermappa_b="1", \$link_szoveg="") függvény az első paraméterben megadott mappában található fájlokat teszi elérhetővé link formájába. Az alapértelmezett gyökérmappát használja, de ez átdefiniálható. A harmadik paraméterrel a linkeknek adhatunk meg alapértelmezett nevet, kulcsot; például: letöltés.

7 Kezelés és általános elvek

7.1 A Métisz kezelése

7.1.1 Szerző

Szerző bárki lehet, írását LDAP-s (4.2) azonosító megadása után tudja feltölteni a rendszerbe. Ha ezzel nem rendelkezik, akkor regisztrációja után az ott megadott felhasználónév és jelszó párossal lehet a rendszerbe belépni és feltölteni a cikket.

Regisztrálni az aktuális folyóirat „Belépés” link alatt megjelenő oldalán lehet a Regisztráció linkre kattintással. Sikeres regisztrációt követően a rendszerbe belépni a „Belépés” menüpont alatt elérhető oldalon, a felhasználónév és a jelszó megadásával lehet. A regisztrációról pontosabban a (7.1.3) fejezetben olvashatunk.

A belépést követően a szerző a következő oldalt látja, hacsak az adott folyóiratban nem definiálták át ezt az oldalt, mert itt akár közvetlenül a cikkek listájához, is ugorhatunk.



1. ábra: Sikeres belépés után

A szerző a beküldött cikkek listáját a „Szerző menüpont” alatti oldalon tekintheti meg. Itt akár nyomon is követheti cikkének állapotát, hogy átesett-e a bírálaton, vagy épp most bírálják, vagy már kiadási állapotban van.

Az új írásának feltöltését az „Új cikk” linkre kattintással kezdeményezheti. A megjelenő oldal egy űrlapot tartalmaz. A mezők első csoportja egy önellenőrző lista, amely a kézirattal

kapcsolatos előírásokat tartalmazza. Minden egyes feltételbe bele kell egyezni, ezt a mondat elején lévő négyzetek bepipálásával lehetséges. Ekkor a szerző vagy feltöltő „igenlő magatartást” tanúsít. Természetesen szinte minden folyóiratban e szövegek - „ellenőrző kérdések” - különböznek.

Kötött ellenőrző lista

- ☐ A kéziratot előzőleg nem adták ki, és más folyóirathoz sincs publikációra beküldve.
- ☐ A kézirat Microsoft Office Word (*.doc, *.docx), Open Office (*.odt) vagy Rich Text Format (*.rtf) formátumú.
- ☐ A szöveg 1.5-ös sorközt és 12-es betűnagyságot használ.
- ☐ A szöveg igazodik a Címleírás és hivatkozási szabályokban vázolt bibliográfiai elvekhez, amelyeket az Információk menüpont alatt talál meg.

2. ábra: Ellenőrző lista

Az űrlap második mezőcsoportjában a szerző saját adatait adhatja meg, ezek a metaadatok, amelyeket a keresés, indexelés és a kezelés során használunk fel. Az adatok bekerülnek a DEA-ba az archiválás után. Az űrlap egyes elemei a profilban beállított értékekkel automatikusan kitöltődnek.

Szerzői adatok

Státusz hozzáadása: Szerző ▼

Szerző:

E-mail:

Ország: ▼

Intézményi hovatartozás:

url:

Életrajzi adatok:

Új elem hozzáadása

3. ábra: Szerzői űrlap

Az „Új elem hozzáadása” gomb megnyomásával további szerzők vagy más közreműködők nevét és adatait adhatja meg. Tetszőleges számú „Új elemek hozzáadására” van lehetőség. Az űrlapon a mezők harmadik csoportja a feltölteni kívánt cikkel kapcsolatos információk

megadására szolgál, ezek természetesen eltérhetnek folyóíratonként, a legtöbb esetben ilyen formában jelenik meg:

4. ábra: Cikk űrlapja

Az űrlapot záró „Kulcsszó” mezőben a szerzőt legalább öt olyan kulcsszó megadására kérjük, amelyek alapján cikkét az internetes keresőmotorok megtalálhatják, illetve az elektronikus folyóirat-szolgáltató adatbázisok indexelhetik. Magyar és angol kulcsszavak megadását is javasoljuk, a cikk nyelve választó mezőben lehetőség van kiválasztani a cikk nyelvét.

Az „**Elmentés**” gomb megnyomása után új oldal jelenik meg, ahol megjelenik egy fájlfeltöltő mező. A „Browse” (Tallózás) gomb megnyomásával kikeresheti a számítógépén a feltölteni kívánt cikket, a „Feltöltésre” kattintva elindul a feltöltés.

5. ábra: Cikk feltöltése

Ha a feltöltés nem sikertelen, akkor erről hibaüzenet értesít, ha nem jelenik meg hibaüzenet, akkor a feltöltés sikeres, ekkor megjelenik a fájlra mutató link. Itt ellenőrizheti, hogy a megfelelő fájlt töltötte fel vagy sem. Ha az újonnan megjelent fájl nevére kattintunk, akkor letölthetjük és ellenőrizhetjük, hogy megfelelő dokumentumot töltöttünk-e fel. Ha nem, akkor a link melletti **-X** linkre kattintva törölheti a kéziratot, majd másikat tölthet fel. Természetesen egyszerre több fájlt is feltölthet - bár ezt nem javasoljuk. Végül az „Elmentés” gomb megnyomásával tud továbblépni a cikk feltöltött adatait összefoglaló oldalra. Itt lehetőség van a feltöltött adatok ellenőrzésére.

Hibás adatok javítását, ha szükséges, a „Cikk adatainak szerkesztése” megnyomásával lehet kezdeményezni. A "Cikk adatainak szerkesztése" és a "kézirat törlése" csak a feltöltés alatt vagy a befejezés után közvetlenül érhetőek el.

7.1.2 Bíráló

A lap főszerkesztője elektronikus levelet küld azoknak, akiket felkért a folyóirat bírálói pozíciójának betöltésére. A levélben megkapja a belépéshez szükséges felhasználónevet és jelszót.

A rendszerbe a Belépés menüpont alatt a megadott felhasználónévvel és jelszóval léphet be.

Belépés után a következő oldalt látja, a táblázatban a bírálendő cikkek találhatóak a rendezésnek megfelelően. Az „Adatlap alatt Bírálom” linkre kattintva tekintheti meg. A cikk adatait és töltheti a bírálendő szöveget.

Bíráló

Aktív cikkek	
Rendezés	
<input type="text"/>	<input type="text"/> kereses
Cím	Adatlapja
z \ z Létrehozva: 2010.02.04. 13:17:10 Utolsó módosítás: 2010.02.04. 13:17:10	Bírálom

6. ábra: Bíráló felülete

A bírálendő szöveg vagy szövegek a „Csatolt Fájlok” csoportban található meg, letöltésük nagyon egyszerű, csupán a fájlra hivatkozó linkre kell kattintani, és lementeni a gépünkre. A bírálatat szövegét, pedig a „Bírálat” csoport szöveges mezőjébe kell beírni. Egyes esetekben a bíráló maga is tölthet fel dokumentumokat, például, ha azonnali javításokat végez a cikkben, vagy bírálati javaslatát dokumentumban is szeretné feltölteni.

The screenshot displays a web interface for document review, organized into three main sections:

- A cikk bírálói változatának feltöltése.** This section contains a file input field with a "Tallózás..." (Browse...) button and a "feltöltés" (Upload) button.
- Csatolt fájlok** (Attached files): This section lists the files attached to the document. It shows a file named **-eredeti_02_05_vpn_pptp_kialakitasa.pdf - X** with a red 'X' icon.
- Bírálat** (Review): This section is for entering the review. It features a large, empty text area for writing. Below the text area, there is a label "A válasz lista:" followed by a dropdown menu and an "Elküldés" (Send) button.

7. ábra: Bíráló felülete 2

A bírálat megírása után a válasz-lista legördülő menüjéből értelemszerűen kiválaszthatjuk az ítéletet leginkább kifejező mondatot.

A rendszer automatikus e-mailt küld, valahányszor újabb cikk bírálatára kéri a szerkesztőst. A Métisz minden bírált cikket, és annak bírálatát elmenti, így ön visszamenőleg is megtekintheti az ítéleteit.

7.1.3 Szerkesztő

A szerkesztői feladatok szerteágazóak, ezért is van a Métiszben két szerkesztői kör. A főszerkesztőhöz érkeznek be az új cikkek és ő is küldi tovább bírálatra, tördelésre és lektorálásra. De ő dönti el, hogy melyik számban, és mikor kerül a cikk kiadásra.

A szerkesztők a beállításoktól függően segíthetik a főszerkesztő munkáját. Az is könnyen megvalósítható, hogy a rovatoknak külön szerkesztőt jelölünk ki, ami azt jelenti, hogy minden joggal rendelkezik, amivel a főszerkesztő az adott rovaton belül.

A szerkesztői felület első és legfontosabb csoportosítása az „Aktív cikkek” csoportja, ebben jelennek meg a szerkesztőnél jelenleg megtalálható cikkek, amelyek valamilyen feladatra várnak. A „Feldolgozás alatt” csoportban pedig azok a cikkek találhatók, amelyek jelenleg valamilyen munkafázisban vesznek részt. Az első oszlop a cikk nevét, létrehozását és az utolsó módosítás dátumát tartalmazza, a következő oszlopban a cikk helyét, hogy melyik felhasználónál található és a milyen szerepkörben.

Aktív cikkek

Rendezés

----- ▾ ----- ▾ kereses

Cím	A cikk jelenleg:	Adatlapja
-----	------------------	-----------

Feldolgozás alatt

Rendezés

----- ▾ ----- ▾ kereses

Cím	A cikk jelenleg:	Adatlapja
magyar cím \ Angol cím Létrehozva: 2010.05.02. 01:35:31 Utolsó módosítás: 2010.05.02. 01:35:31	Dudás Csaba(Bíráló)	Adatlapja

8. ábra: Szerkesztői felülete cikk lista

Az „Adatlapja” linkre kattintva a cikk szerkesztői felületéhez és adataihoz navigálhatunk. Ezen a felületen lehet a cikket tovább küldeni bírálóknak, tördelőknek, és egyéb munkatársaknak. Ha kiválasztjuk a megfelelő szerepkört, akkor egy újabb űrlap bukkan fel.

Cikk műveletek

Válasszon feladatkört:

tovább

Cikk útja

Szerző
Szerkesztő
Bíráló
Tördelő
Korrektor
Főszerkesztő

9. ábra: Cikk elküldése

Ezen az űrlapon választhatjuk ki azt a felhasználót, akinek el szeretnénk küldeni a cikket. Itt csak azoknak a felhasználóknak a nevei láthatóak, akik rendelkeznek a megfelelő felkéréssel.

Cikk műveletek

Tejlesnév keresése

Választható felhasználók:

Elküldés

Válasszon feladatkört:

tovább

/
Dudás Csaba/
Pete László/

10. ábra: Személy kiválasztása

Mivel itt többen is szerepelhetnek ebben a legördülő menüben, mint amennyi kényelmesen kezelhető, így lehetőséget ad a rendszer a „Teljesnév” keresésre. Ebben az esetben, az erre kijelölt mezőbe beírjuk a keresett nevet, vagy annak részletét, majd egy Enter megnyomásával a legördülő lista elemei leszűkülnek azokra az elemekre, amelyek a keresés találati listájában viszajönnek. Későbbi fejlesztésünk során szeretnénk, ha itt nem csak a felhasználó neve, hanem egyéb, az azonosításra szolgáló adatok is megjelennének.

Ha kiválasztjuk a megfelelő nevet, akkor ismét egy űrlap jelenik meg, ahol az előre deklarált generált levelet láthatjuk, amit elektronikus levélként fog megkapni, továbbá a cikk azonnal hozzá kerül.

Cikk műveletek

Tejlesnév keresése

Választható felhasználók: Dudás Csaba/

Level_targya:

Bírálóí felkérés

Tisztelt Dudás Csaba!

Fel kérem önt a magyar cím \ Angol cím nevű cikk bírálására.

A cikk szerzője: Dudás Csaba

Elküldés

11. ábra: Felkérés elküldése

A „Cikk útja” információs csoport azt taglalja, hogy a cikk milyen utat járt be, és jelenleg kinél található. Ha pedig a cikkel nem történik semmi, akkor visszavonhatjuk a felkérést, ekkor egy generált levél jelenik meg, amelyet elfogadva a cikket elvesszük a felhasználótól, és értesítjük e-mailban.

Cikk útja			
Teljes név:	Szerepkörben	Megkapta	Leadta
Dudás Csaba	Szerkesztő	2010.05.02. 01:35:31	2010.05.02. 01:38:59
Dudás Csaba	Szerző	2010.05.02. 01:35:31	2010.05.02. 01:37:50
Dudás Csaba	Bíráló	2010.05.02. 01:38:59	Visszavonom

12. ábra: Cikk útja

A következő csoport teljes információt nyújt a cikkről, itt jelennek meg a bírálatok is. Ha szeretnénk elolvasni a bírálatot, akkor rá kell kattintani a bírálat első mondatára, és automatikusan betöltődik a teljes bírálat.

Cikk adatai.	
Létrehozva:	2010.05.02. 01:35:31
Utolsó módosítás.:	2010.05.02. 01:35:31
A cikk jelenleg:	Dudás Csaba(Bíráló)
Magyar cím:	magyar cím
Angol cím:	Angol cím
Rovat	
Folyóirat számban	
Absztrakt	Absztrakt
Kulcsszó	Kulcs szavak
Szerző	
Státusz hozzáadása	Szerző
Szerző	Dudás Csaba
E-mail	csdudas@gmail.com
Ország	HU
Intézményi hovatartozás	Debreceni Egyetem
Életrajzi adatok	teszt

13. ábra: Cikk adatai

7.1.4 Regisztráció

A regisztráció során pár fontos adatot kérünk a felhasználótól. ezeket az adatokat később a „Profil” menüpontban módosíthatja (7.1.5). A mezők kitöltése nem csupán az azonosítást segítik, de egy új cikk feltöltésénél ezek az információk azonnal beíródnak. **(Hiba! A hivatkozási forrás nem található.)**

The registration form is divided into three main sections:

- Rendszer adatok:**
 - Felhasználónév:
 - Új jelszó:
 - Új jelszó ismét:
- Személyes adatok:**
 - Teljes név:
 - Születési idő:
 - E-mail cím:
 - Telefonszám:
- Életrajzi adatok:**
 - Intézményi hovatartozás:
 - Irányítószám:
 - Település:
 - Uta:
 - Házszám:

A large text area for biographical details is located below the personal data fields. At the bottom right, there is a button labeled "Adatok elmentése".

14. ábra: Regisztrációs űrlap

7.1.5 Profil

A „Profil” linkre kattintva megjelenik a profil oldalunk, ahol módosíthatjuk személyes adatainkat és lecserélhetjük jelszavunkat. Nagyon fontos, hogy ha LDAP (4.2) felhasználó névvel és jelszóval jelentkezik be, akkor a jelszó ebben a rendszerben nem módosítható.

7.1.6 Rendszerfordítás, szó/kifejezés kezelés

A rendszer programozása közben bármilyen kiírást, kifejezést, szót, mondatot, linket egy saját azonosítóval láttunk el, és a programkódban ezt a kulcsot helyeztük el. Ezzel biztosítjuk, hogy bármilyen nyelvre egyszerűen és könnyen lefordíthassuk a kezelőfelületet, a programkód átírása, vagy módosítása nélkül.

Egy új folyóirat esetében alapértelmezett szavakkal és kifejezéssel töltjük fel az adatbázist, a szavakat az erre jogosultak módosíthatják. A „Fordítási mód:” szöveg melletti pipa bejelölésével aktiváljuk a fordítást segítő felület. Itt értelemszerűen kiválasztjuk a kívánt nyelveket, majd az elmentés gombra kattintunk, ekkor a legördülő menüben csak a lefordíthatlan kifejezések jelennek meg.

Fordítási mód: ☒ Erről: ---- Erre: ---- Elmentés

Szó/kifejezés: --új kulcs--

Keresés:

Kulcs:

Szó/kifejezés:

Nyelv: ---- Elmentés

15. ábra: Fordító felület

A kifejezések módosítása is nagyon egyszerű, hiszen csak a legördülő menüben kijelöljük a módosítandó kifejezés kulcsát, majd a lenti mezőben a változtatásokat végre hajthatjuk. Fontos, hogy a „Nyelv” legördülő menüben azt a nyelvet válasszuk ki, amelybe szeretnénk menteni.

8 Várható fejlesztések

Kitűzött cél a rendszer modularitásának növelése, a programkódok átláthatóbbá tétele, új függvények és eljárások készítése, amelyek könnyítik a további fejlesztést.

A jogosultsági rendszer kezelőfelületét mindenképpen tovább kell fejleszteni, és egy jobb fő-rendszergazdai felületet kell készíteni, amely segítségével könnyebben és hatékonyabban irányíthatjuk és karbantarthatjuk az egymástól független folyóiratokat.

Nagy fejlődési irány a cikkek megjelenésére, tördelésére szánt energia csökkentését segítő megoldások, jelenleg egy könnyű Word-szerű szerkesztőfelület áll a tördelők rendelkezésére. A távlati cél az, hogy a beküldött cikket a rendszer automatikusan megformázza és legenerálja a kívánt kimeneteket, de legalábbis a HTML oldalt.

A rendszer megjelenési felülete mindenképpen átírássra szorul, a módosítás célja, hogy minél keresőoptimalizáltabban jelenjenek meg a cikkek. A cikkek megjelenítő felületét megfelelő metaadatokkal kell feltölteni.

Fontos és nemsokára bevezetett fejlesztés az a lehetőség, hogy a cikkeket a szám kiadása előtt is lehetőség legyen kiadni, ekkor a közvetlen URL-lel, amely a cikk élete során nem változik, vagy egy egyedi azonosítóköddal lehet a cikkekre hivatkozni. Miután kiadásra kerül a szám, a cikkhez hozzárendelődik az oldalszám és a folyóirat szám.

A rendszerben jelenleg csak böngészni lehet a cikkeket, a keresés megoldására több alternatíva is kínálkozik: lehetséges, hogy a DEA keresője beépül a rendszerbe, ez a megoldás problémákat vet fel, de egyben sok kérdésre válaszolna, és a rendszerek összedolgozását és kezelőfelületük megegyezését segítené. Elképzelhető azonban az is, hogy a Google kereső API-ját használjuk fel.

9 Összefoglalás

Céлом, hogy a rendszer fejlesztésével olyan állapotot érjünk el, amely során a program telepítése egyértelmű és átlátható, könnyen végrehajtható, a folyóiratok létrehozása és beállításai egyszerűen alkalmazhatóak, a felhasználói felület magától érthetődő. A távlati gondolatok jelenleg azt mutatják, hogy a Métisz egy nyílt elérésű rendszerre fejlődik, így bárki letöltheti és felhasználhatja a rendszert saját folyóirat megjelentetésére vagy kezelésére.

A dolgozatomban bemutattam gondolataimat a nyílt rendszerekről, elérésükről és hozzáférésükről, az OJS használatáról, a problémákról, a kódolás alapelveiről, azokról a rendszerekről, amelyek felhasználása szinte elengedhetetlen egy komoly könyvtári rendszernek. Ismertettem a főbb függvényeket, azok visszatérési értékét és függőségét.

A rendszer kezelésének bemutatását a lehető legrészletesebben mutattam be: minden folyóirathoz tartozik egy információs lap, és egy súgó. A súgó tartalma szintén a rendszer kezelőfelületének bemutatása, de az nem ilyen részletes, és nem tartalmazza azokat a megoldásokat, amelyek a rendszerek egyediségét, könnyítését valósítják meg, de itt kitértem ezekre a megoldásokra is.

Mint látjuk sok lehetőség és fejlesztés vár még a rendszerre, remélem ezek többségét lehetőségem lesz megvalósítani, és ezzel egy olyan rendszert hozhatok létre, amely könnyű, egyszerű, és segíti a magyarországi nyílt tudományos elektronikus folyóiratok megjelenítését, és számának növelését.

Természetesen ez a rendszer nem minden problémára megoldás, de egy kezdeményezés, amelyhez reményünk szerint egyre többen csatlakoznak, amely szükséges a mai társadalmi és gazdasági helyzetben. A munka legnagyobb része még előttünk áll, a fejlesztések célja, hogy megkönnyítse a felhasználhatóságot és a kezelhetőséget.

10 Idézett forrásmunkák

ALA. (1989). *Final Report of the ALA Presidential Committee on Information Literacy*.

Forrás: <http://www.ala.org/ala/acrl/acrlpubs/whitepapers/presidential.htm>

Björk, B.-C. (2004). *Open access to scientific publications - an analysis of the barriers to change?* Letöltés dátuma: 2010. 04 23, forrás: informationr.net: <http://informationr.net/ir/9-2/paper170.html>

Corrado, E. M. (2005). *Global Access to Indian Research: Indian STM Journals Online*.

Letöltés dátuma: 2010. 04 23, forrás: istl.org: <http://www.istl.org/05-spring/article3.html>

DEA. (dátum nélk.). Forrás: [dea.hu: http://ganymedes.lib.unideb.hu:8080/dea/](http://ganymedes.lib.unideb.hu:8080/dea/)

Harnad, S. (2005). *The Implementation of the Berlin Declaration on Open Access*. Letöltés

dátuma: 2010. 04 23, forrás: [dlib.org: http://www.dlib.org/dlib/march05/harnad/03harnad.html](http://www.dlib.org/dlib/march05/harnad/03harnad.html)

Métisz. (2010). Letöltés dátuma: 2010. 04 23, forrás: [wikipedia.org: http://hu.wikipedia.org/wiki/M%C3%A9tisz](http://hu.wikipedia.org/wiki/M%C3%A9tisz)

Nicholas, D. R. (2005.). Open Access Publishing: The Evidence from the Authors. *Journal of Academic Librarianship*. vol. 31., no. 3. , 179–181.

PHP. (dátum nélk.). Forrás: [wikipedia.org: http://hu.wikipedia.org/wiki/PHP](http://hu.wikipedia.org/wiki/PHP)

SOAP. (dátum nélk.). Forrás: [wikipedia.org: http://hu.wikipedia.org/wiki/SOAP](http://hu.wikipedia.org/wiki/SOAP)

Tibor, K. (2007). *A nyílt hozzáférés*. Letöltés dátuma: 2010. 04 23, forrás: [tankonyvtar.hu: http://www.tankonyvtar.hu/konyvtartudomany/virtualis-elektronikus-080906-66](http://www.tankonyvtar.hu/konyvtartudomany/virtualis-elektronikus-080906-66)

Tibor, K. (2007). *A nyílt rendszerek*. Letöltés dátuma: 2010. 04 23, forrás: [tankonyvtar.hu: http://www.tankonyvtar.hu/konyvtartudomany/virtualis-elektronikus-080906-64](http://www.tankonyvtar.hu/konyvtartudomany/virtualis-elektronikus-080906-64)

Tibor, K. (2007). *A nyitott szabványok*. Letöltés dátuma: 2010. 04 23, forrás: [tankonyvtar.hu: http://www.tankonyvtar.hu/konyvtartudomany/virtualis-elektronikus-080906-65](http://www.tankonyvtar.hu/konyvtartudomany/virtualis-elektronikus-080906-65)

Zoltán, P. (2003. 04 30). *LDAP-ról pár szóban*. Letöltés dátuma: 2010. 04 23, forrás: [padre.web.elte.hu: http://padre.web.elte.hu/ldap.html](http://padre.web.elte.hu/ldap.html)

Zsolt, B. (2003). *Nyílt Hozzáférés Kezdeményezés (Open Access Initiative) – Kitekintés és körkép.* Letöltés dátuma: 2010. 04 23, forrás: tmt.omikk.bme.hu :
http://tmt.omikk.bme.hu/show_news.html?id=2093&issue_id=66