

Szakdolgozat

Csehy Ladislav

Debrecen
2009

Debreceni Egyetem Informatikai Kar

Alfresco

Egy újabb nagyágyú a piacon?

Témavezető:
Dr. habil. Boda István
Tanszékvezető

Készítette:
Csehy Ladislav
Informatikus könyvtáros

Debrecen
2009

Tartalomjegyzék:

Tartalomjegyzék:	1
Bevezetés	5
Alfresco	6
Letöltés és telepítés	7
Alfresco Community Labs.....	8
Alfresco Enterprise Edition	8
Telepítés.....	9
Első lépések. Ismerkedés a termékkel	11
ECM rendszerek birtoklási költségei.....	11
EMC/Dokumentum	13
OpenText	13
SharePoint.....	14
Alfresco	14
ECM eszközök ára	15
Oracle Database and Application Server.....	15
Microsoft Database.....	16
Microsoft Windows Server.....	16
MySQL Database	16
Red Hat Application Stack	16
Alfresco	17
Összefoglalás	17
Nyíltság fogalma.....	18
Alfresco felépítése	21
XML	24
Formai elvárások	25
Felhasználási területei az XML-nek:.....	26
Értelmezők / validator	28
Példák.....	29
Alfresco rekordkezelése	36
Összefoglalás:.....	40
Alfresco keresési mechanizmusa.....	40
Nyílt keresés	43
Keresés és biztonság.....	46
Enterprise Social Software	46
A Social Software legfontosabb hatásai	47
Alfresco Share	52
Egyéni műszerfal létrehozása	56
Együttműködési oldal készítése.....	57
Együttműködési oldal beállítása.....	57
Oldal elemek.....	58
Share oldal elemeinek a legfontosabb sajátosságai	59
Share használata.....	61
Engedély modellek referenciája, egy kis rálátás a jogosultságokra	63

Egyszerű példa engedélyekre	64
Dokumentum kezelés	67
Biztonság.....	69
Alfresco dokumentumkezelése	73
Összegzés.....	77
Képjegyzék	78
Hivatkozásjegyzék:.....	79
Köszönetnyilvánítás.....	80

Bevezetés

Az 1980-as évek elejére tehető az, amikor egyesek a papír-alapú és hasonló dokumentumok kezelésére számítógépes rendszereket kezdtek el létrehozni. Hamar rájöttek, hogy itt nem csak az irodában felhalmozott sok irat kezelésére lehetne használni, hanem bármilyen dokumentumtípusra, ami egy ilyen intézetben előfordul. Később egy második rendszert alkottak meg, mely az elektronikus dokumentumok kezelésére lett hivatott. A legelső elektronikus dokumentumkezelők (electronic document management, ECM) csak a cég saját dokumentumait, vagy esetleg egy-két más fájl formátumot is támogattak. Ezek a rendszerek képesek voltak faxok és nyomtatványok feldolgozására, képként való mentésére az adattárterületen, biztonság és a gyors visszakeresés céljából.

Majd később ezekből a rendszerekből, olyan dokumentumkezelők jöttek létre, melyek már képesek voltak az összes fájl kezelésére, és hálózaton való mentésére. A rendszerek tudása folyamatosan nőtt, felölelte az összes elektronikus dokumentum formátumot, együttműködési eszközöket építettek beléjük, nagyobb biztonsági követelmények léptek életbe és naplózási lehetőségek biztosították, hogy mindenről, ami használat közben történt feljegyzés készüljön.

A mai „rohanó” világban nincs helye tévedésnek. A tévedés sokba kerülhet, melyet a most érő gazdasági válság csak fokoz. A cégeknek át kell fogalmazniuk a stratégiájukat, ha továbbra is fent szeretnének maradni ebben a gazdaságilag instabil helyzetben. A dokumentumok és a tartalom lételeme a kormányoknak, szervezeteknek, cégeknek és lakosoknak. Arra használják őket, hogy tudást, információt tároljanak (benne), tevékenységeket hangoljanak és eredményeket hasonlítsanak össze, és az ebből nyert információkat a külvilágnak szolgáltatassák. Ehhez szükség van egy olyan rendszerre, ami ezt biztosítja, sőt ezt merőben túlteljesíti. Erre a megoldást a vállalati tartalomkezelők (Enterprise Content Management, ECM) jelentik. Mindenki, aki valaha is dolgozott számítógéppel, és ismeri valamilyen szinten, az tudja, hogy a tartalom, legyen az bármi - a legtitkosabb katonai fejlesztésektől kezdve a cégek megjelentetett kiadványain keresztül egészen egy család személyes képalbumáig - a különböző fájlok mind mappákban vannak rendezve. Ez a lehető legegyszerűbb módja, hogy különböző tartalmakat választhassunk el egymástól. Példának

okáért szerintem a család képalbuma teljes mértékben megfelelő, de mi történik akkor, ha ez a fajta rendezés kevésnek bizonyul?

Alfresco

Az Alfresco Software cég terméke, melyet szakemberek csapata fejleszt, akiknek vezető szerepük volt a Documentum¹ és az Interwoven² programok elkészítésében. A tagok között megtaláljuk:

- John Newton-t aki a társszerzője a Dokumentumnak
- John Powell
- Dr. Ian Howells
- Matt Asay

És további vezető személyek Dokumentumból, FileNet³-ből, OpenText⁴-ből, Interwoven-ből és Vignette⁵-ből, növelik a cég nívóját.

A cég 2005-ben alakult és, azóta az egyik vezető szolgáltatóvá nőtte ki magát. Ennek eredménye, hogy 2005-ben az Alfresco megnyeri az Emerging Elite Award díját, majd 2006-ban már három díjat nyert, köztük a legrangosabb a Red Herring cím. 2007-ben már kilenc címet tudhat magáénak, ilyen az EContent 100 díj és 2008-ban hat díjat kap, többek között az Audemars Piguet and European Tech Tour Association-tól a Next Gem díjat kapta, a legjobb új cég elismeréseként. Használói köre ezáltal, szintén sokrétű, ide sorolható:

- Activision
- Air Force (France)
- American Stock Exchange
- EA
- Harvard Business
- Los Angeles Times

¹ <http://www.documentum.com/>

² <http://www.interwoven.com/>

³ <http://www.01-ibm.com/software/data/content-management/filenet-content-manager/>

⁴ <http://www.opentext.com/>

⁵ <http://vignette.com/>

- Massachusetts Institute of Technology
- NASA

Ilyen gazdag felhasználói körnek az igényeit kielégíteni nagy kihívás, amivel szemben az Alfresco a legújabb technológiákat vonultatja fel. Szó lesz az Alfresco dokumentum kezelőjéről (Alfresco Document Management, DM), a webes tartalom kezelőről (Alfresco Web Content Management, WCM), a rekord kezelőről (Alfresco Record Management, RM). Itt szeretném megjegyezni, hogy sajnos nem tudom leírni a saját tapasztalataimat, ugyanis ahogy kiderült, gondok vannak a 3.X változatoknál, a modulok bejátszásának a kompatibilitásával erről ott kicsit pontosabban. De természetesen, aminek utána néztem a rekord kezelővel kapcsolatban, azt lehetőségeim szerint ismertetem. Továbbá megemlítem a modulok megvalósításához szükséges programokat, lehetőségeit. Keresési lehetőségeket, mint nyílt keresést (OpenSearch, OS). Említés szintjén szó lesz a Web 2.0-ról, és hogy ez miben érinti az ECM világot, vagyis a társadalmi szoftver meghatározásáról (Enterprise Social Software)-.

Írni szeretnék az XML-ről, ami nélkül nem beszélhetnék az előbb említett eszközökről.

Munkámban a végső felhasználó lehetőségeit szeretném bemutatni. Tisztában vagyok vele, hogy közel sincs meg hozzá a tudásom, hogy egy ilyen volumenű program programkódú bemutatását elvégezzem, illetve – sajnos - ahhoz sincs, hogy az egyes felhasznált technológiákat bemutassam. Igazából itt a lehetőségekről beszélnék, arról, amit nyújt az Alfresco és nem arról, hogy ezt hogy, milyen módon éri el (csak említési szinten).

Letöltés és telepítés

Az Alfresco két lehetőséget nyújt, hogy elérjük a termékeit. Az egyik a vállalatok számára tervezett Alfresco Enterprise Edition (fizetős) és az Alfresco Community Labs. Mindkét esetben a 3.1-es verzió a legfrissebb most.

Alfresco Community Labs

Alfresco által nem támogatott fejlesztésre szánt program csomagja. Napi szinten frissíthető, és a legújabb funkciók is használhatók benne. Arra szolgál, mint egy „laboratórium”, kutatásra, ahol új fejlesztéseket, innovációkat lehet eszközölni. Ennek köszönhetően sok újítás, fejlesztés érkezik be nap mint nap az Alfresco-hoz, és amiatt, hogy ilyen sokan fejlesztik, nagyon látványosan tud fejlődni, hogy mindig a legjobb lehessen a piacon. E köré építette fel az Alfresco Community platformot, ahol az egyes külső-belső fejlesztők megoszthatják az új munkáikat, megvitathatják a nézeteiket, beszélgethetnek stb. Community Lab jellemzői:

- 100% nyílt forrású
- Naponta frissülő SVN⁶ hozzáféréssel
- Se Alfresco se más hivatalos partner támogatás
- Semmiféle méretezhetőségi teszt nem áll rendelkezésre

Alfresco Enterprise Edition

Állandó fejlesztésből (Labs) egy stabil, megbízható programig, ez az Enterprise Edition. Használatra kész verzió, ami az Alfresco Software Inc teljes támogatását élvezi. A forráskódok 95%-át az Alfresco csapata írta, ami teljesen egyedivé, és a piacon a legmodernebb ECM rendszerré teszi, hála a nyílt forrású programozásnak. Ez a változat olyan szervezeteknek, cégeknek, intézményeknek szeretne megfelelni, amelyeknek kereskedelmi szintű szolgáltatási egyezményekre van szükségük (Service Level Agreements, SLAs). Enterprise Edition jellemzői:

- 100% nyílt forrású
- Kereskedelmi szintű támogatás
- Hitelesített méretezés

Az ECM kiszolgálására épült Alfresco vállalati hálózatot (Alfresco Enterprise Network) úgy tervezték, hogy pénzt, időt és emberi erőforrást spóroljon meg, még hozzá úgy, hogy szinte azonnal használatba vehető és teljes körű támogatást bizto-

⁶ Subversion, <http://www.wikipedia.org/wiki/Subversion>

sít. Hogy ez gördülékenyen történjen kereskedelmi szintű szolgáltatást nyújt, ami az alábbiakból áll össze:

- Feljavítások, bug javítások és karbantartás
- Garancia és termékjótállás
- Stabil jogtisztta termék
- Támogatás / segítségnyújtás
- Teljesen tudásbázist a termékekről

Összehasonlítást és további részletes információkat a termékekről a <http://www.alfresco.com/products/networks/> weboldalon találunk meg. Továbbiakban a Labs-al nem foglalkozom, csakis az Enterprise Edition-nel. Ahhoz hogy letölthessük a programot, először is be kell magunkat regisztrálni az Alfresco Content Community⁷-ba. Itt elkérik az adatainkat, és a jóváhagyás után kapunk egy értesítő e-mailt az általunk megadott felhasználó névvel és az általuk automatikusan generált kóddal, amivel be tudunk lépni. Ha kiválasztottuk, hogy milyen operációs rendszerre kívánjuk telepíteni, akkor ezután szüksége lesz a felhasználói nevünkre és jelszavunkra, majd szükséges kiválasztani, milyen webszervert használjon, milyen Office csomagot, és azt, hogy milyen adatbázis legyen hozzárendelve. Ezután már csak a letöltés gombra kell kattintani, és már kezdődhet is a letöltés.

Telepítés

Miután letöltöttük a kívánt Alfresco terméket indulhat a telepítés. Én a WCM Enterprise Edition trial verzióját töltöttem le, Tomcat szerverrel, MS Office támogatással és az alap Derby adatbázist választottam, XP operációs rendszer alatt. A WCM Enterprise Edition annyit jelent, hogy egy teljesen működőképes Alfresco Enterprise Network programot kaptam, és ezen belül a tartalomkezelőt és a webes tartalomkezelőt tudom teljesen kihasználni, de sajnos csak 30 napig. A telepítést elindítva kellemes meglepetésben lettem részem, ugyanis magyar menüvel tudtam a telepítés varázsló lépéseit megtenni. A telepítés elején felkínálta a többi nyelv mellett a magyart is, ami automatikusan alapbeállítás volt. Ebből arra következtetek, hogy a Windows nyelvét felismeri. (és ha van olyan nyelvi lehetőség, akkor azt ki-

⁷ <http://www.alfresco.com/community/register/>

nálja fel.) Tovább kattintva egy tipikus (typical) és egy saját (custom) beállítások közül lehet választani. A tipikus beállítás a C:\Alfresco mappába telepíti a Tomcat szerverrel és HSQL⁸ (Hyper Structured Query Language) adatbázissal együtt, amit csakis tesztelésre és értékelésekre ajánlanak. Továbbá bejátszásra kerül a WCM a java (Java Development Kit, JDK), az OpenOffice (fájl típusok közti átalakítások miatt van) és a SharePoint⁹ protokoll. Az egyedi telepítés ugyanez, csak itt kiválaszthatjuk hogy, mely részeket szeretnénk esetleg kihagyni. A következő oldalon a meghajtó kiválasztása történik majd a telepítés szakasza elindul feltéve, ha van JDK 1.5 és OpenOffice telepítve a gépünkön, ha nincs akkor ezt pótolni kell, de ha szeretnénk automatikus letölti, és telepíti helyettünk. Ha nem a Derby adatbázis szervert választottuk, hanem a felkínált MySQL szervert, akkor a telepítés feltétele, hogy már telepítve legyen. Ilyenkor egy plusz oldal jelenik meg, ahol a MySQL adatbázisunk adatait kell beírunk. Ezután megtörténik a telepítés.



1. kép: telepítés során 4x kaptam 2 különböző gépen ezt a hibát

A befejezés után rögtön indítható is, semmilyen külön beállítást a továbbiakban nem igényel. Itt meg kell említenem, hogy ez egy alapbeállítást szolgáltat, ahol elérhetünk minden lehetőségét, de természetesen még nagyon sok beállítást és finomítást kell rajta végeznünk, hogy megfeleljen az elvárásainknak. Igazából mi

⁸ <http://en.wikipedia.org/wiki/HSQLDB> 2009.05.07.

⁹ <http://en.wikipedia.org/wiki/SharePoint> 2009.04.28.

csak egy „csupasz” programot kapunk, ami tudásának egy kis részét szolgáltatja. Igazán az egyedi beállítások után derül ki az, hogy mit tud, amihez kis hangolás, vagy erőszakos módosítás vezet.

Első lépések. Ismerkedés a termékkel

Az Alfresco webszerverének elindításához két lehetőségünk van, vagy a start menüből, vagy a gyökérkönyvtárából duplaklikkel az `alf.start.bat`-ra kattintunk. Pár másodperc alatt felfut a Tomcat szerverünk, és vele együtt betöltődik az Alfresco működéséhez szükséges adatok. Ezután a böngészőnk felhasználói felületként arra fog szolgálni, hogy a tartalomkezelővel kommunikálni tudjunk. Itt vagy beírjuk az URL-t <http://localhost:8080/alfresco> amivel el tudjuk érni, vagy a start menüből kényelmesen elindíthatjuk. A szerver leállításához az `alf.stop.bat` fájl szolgál, avagy szintén a start menüből a Stop Alfresco Server parancs. Ha a Tomcat szerverünk felfutott, akkor indíthatjuk a webböngészőnket, ahol a bejelentkezési felület jelenik meg először. Alap felhasználói nevünk és jelszavunk is az „admin”.

Első belépésemkor, még szinte semmit sem tudtam a rendszerről, nehezen boldogultam Sok mindenem nem tudtam kiigazodni, meg kellett szoknom ezt a felületet Nem is amiatt volt számomra nehéz, mert rendezetlen, vagy nagyon sok információt tartalmazna, hanem azért, mert egy teljesen ismeretlen programot kellett megismernem, amit igazán csak a (hozzáértő) szakemberek tudnak értékelni. Természetesen a célom az, az volt, hogy minél jobban elsajátítsam, megértssem, megismerjem ezt a programot.

ECM rendszerek birtoklási költségei

Ennek a résznek semmi köze sincs a program futásához, nem tartalmaz egyetlen olyan információt sem, amely a program futásával, kezelésével, beállításával foglalkozna, de mégis én – személy szerint - nagy szerepet tulajdonítok ennek. Jelen gazdasági helyzetben mindenhol felmerül a kivitelezés, a használat, a jogviszony meghosszabbításának ára, ami esetenként nagyon drága lehet. Épp most, amikor az anyagi biztonság mindenütt kivételesen fontossá vált, úgy érzem, hogy

nem hagyhatom ki ezt a témát sem, mert merőben befolyásolja a használót vagy vevőt, legyen az bárki, aki eziránt potenciálisan érdeklődik.

Arról lesz szó, hogy hogyan tudjuk elkerülni a más cégek formátumából adódó kötöttségeket, és arról hogy, hogyan tudjuk nagymértékben csökkenteni a kiadásainkat a nyitott forráskódú program segítségével. Összehasonlítottam a piacon lévő vezető rendszereket. Így az EMC/Documentum, OpenText, SharePoint és az Alfresco által kínált árjegyzékeket, evvel biztosítva a rálátást a piacon megtalálható különböző ECM rendszerek követelményeire, és ebből adódó díjszabásokra. Általában egy ECM rendszerről úgy vélekednek, hogy nagyon drága maga a licenszelése (használata), továbbá a hardware és szoftver felé nyújtott követelmények, és a méretezés problémája, az, hogy hány emberre is kell méretezni a programot.

Hosszú időn keresztül az ECM ipar magas árú és rendkívül bonyolult modellezési technikára épített, ahol a szolgáltató irányította a vásárlót. Ebben a helyzetben az ECM rendszerek árazása is olyan sokrétű volt, mint a rendszer nyújtotta szolgáltatásoké. Minél többet tudott annál drágább volt, ahol nagyon sok lehetőség közül lehetett választani, ott a kiválasztott lehetőségek száma határozta meg a végső árat. A vásárlók nincsenek tisztában azzal, hogy mit is vesznek meg, mit kellene, hogy tartalmazzon az ő összeállításuk, az általuk szolgáltatni kívánt minőség elérésére. Rengeteg ilyen „rejtett csapda” van, ami csak a szolgáltatónak jelent hasznot, mert evvel is csak pénzhez jut. Ezért szükséges megismerni a piacon lévő lehetséges ECM rendszereket, és egy pontos áttekintést tenni arról, hogy mit is kapunk a pénzünkért.

Szerencsére van egy új lehetőség. A világ legnagyobb weboldalai, már a nyitott forrású programokat használják. Ezek az oldalak a Web 2.0 nyújtotta előnyöket használták ki, saját piaci helyzetüknek növelésére. A Web 2.0 egy sor új lehetőséget nyújt a tartalom elérésére és összekapcsolására. Ez megváltoztatta az ECM rendszerek gazdasági nézeteit, megbízhatóságát, elérhetőségét és méretezhetőségét. A nyílt forrású kódot úgy lehetne jellemezni, hogy „többet kevesebbért”, ami arra koncentrálna, hogy: olcsóbb és egyszerűbb legyen, és az ügyfél felé még több lehetőséget nyújtson.

A következőkben szereplő összes adatok mind 1000 és 100 felhasználóra méretezett beállítás, ahol a teljesség kedvéért a termékek neve, mennyisége is fel van tüntetve. A közölt adatok nyilvánosak, és a rendszerek mindegyike képes a következő feladatokra:

- MS Office integráció
- együttműködés
- tartalomkezelés
- munkafolyam irányítás vagy üzleti folyamatirányítás
- típuskonverzió, fordításkezelés
- keresés

EMC/Dokumentum

Több mint 1 000 lehetőséggel büszkélkedhet. Árazási szempontjai:

- tartalomkezelő szerver
- felhasználónkénti árazás
- többlet pénz kliens típustól függően
- többlet pénz folyamatirányításért
- többlet pénz tartalmi típuskonverzióként
- többlet pénz arra, hogy PDF-t jegyzetekkel lássa el

1000 emberre számolva az első évi költség 863 938 dollár, míg 100 emberre számolva 129 079 dollár. Minden egyes plusz szolgáltatás ára benne van.

OpenText

Több mint háromezer funkció. Árazási szempontjai:

- tartalomkezelő szerver
- modulonkénti árazás, százalékban az alap termék árából
- minden egyes szerverért és modulért
- olcsóbb ár a csakis olvasó-felhasználóért, de drágább árazás a teljes jogkörű használóért

Itt 1 000 emberért az első évben 637 304 dollárt kell fizetni, míg 100 emberért

196 754 dollárt.

SharePoint

Elengedhetetlen tudni, hogy ez a rendszer csakis Windows alatt fut, ami hosszú távú lekötést kíván meg a Microsoft felé. Tipikus rendszer tartalmaz:

- Explorert
- SharePoint Portált
- Office csomagot
- SharePoint Designer
- Webes részeket, mint ASP, .NET, C# API
- SQL Server
- Internet Information Server
- Windows Server 2003 és újabb

Ezért ez a rendszer beállítás tiltja a következő dolgokat:

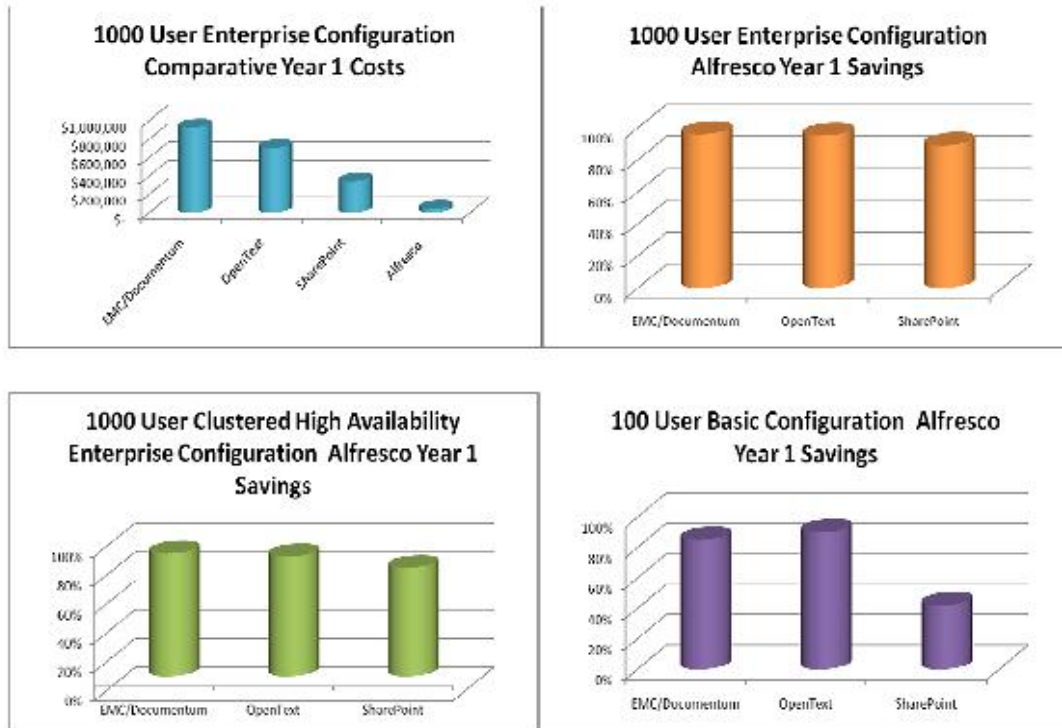
- Linux, Unix
- Oracle, DB2, MySQL
- J2EE, JBoss
- PHP, Java, Adobe
- Továbbá kevesebb funkcionalitás a legelterjedtebb böngészőkkel, kivétel Internet Explorer

1 000 emberre az első éves szerződés ára 318 738 dollár, ahol meg kell említenem hogy, két szerver ajánlatos. 100 emberre pedig az ára 24 669 dollár, ami „Open Value Program” konfiguráció néven ismeretes. Itt már egy szerver is elég.

Alfresco

Itt minden termék egy előfizetésben összpontosul és nincs felhasználónkénti CAL (kliens hozzáférési engedély) árszabás. 1 000 emberre első évben az ára: 33 500 dollár, ami két szervert tartalmaz, és 100 emberre: 18 500 dollár/év, itt egy szerver van. Továbbá lehetőség van még csoportos/klaszterezett magas-hozzáférhetőségű

rendszer megvásárlására is, ami az első évben: 46 250 dollár ezer főre. Itt a teszt szerverből két darab van.



2. kép: Documentum, OpenText és Share Point árazásának összehasonlítása Alfresco-éval

ECM eszközök ára

Egy tipikus ECM rendszer a verem tetején fut, a legfelsőbb szinten, amihez elengedhetetlen: az adatbázis szerver, alkalmazás szerver és egy Operációs rendszer. Egyes ECM rendszerek megkövetelik az alkalmazás szervert, mások belesomagolják, megint másoknak nincs rá szükségük.

Oracle Database and Application Server

Egy Oracle adatbázisnak 1 000 emberre számolva az első éves ára: 59 950 dollár, és ha szeretnénk az alkalmazást szervert is, akkor 88 450 dollárba kerülne. 100 emberrel számolva az ára első évben: 21 350 dollár lenne, és ha szeretnénk a alkalmazás szervert is akkor 33 550 dollárunkba kerülne.

Microsoft Database

1 000 emberre két SQL adatbázis szükséges, amely ára első évben: 17 210 dollár.

100 emberre egy SQL adatbázis szükséges, amely ára első évben: 8 605 dollár.

Microsoft Windows Server

1 000 emberre a konfiguráció három szerveret tartalmaz, aminek az első éves költsége 3 201 dollár.

100 emberre a konfiguráció két szerveret kíván meg, aminek az első éves költsége 2 176 dollár.

MySQL Database

Itt bérleteket kell vásárolnunk, melyek árát szerverszám/év-ként számolják ki. Így

1 000 emberre egy éves használati díj ára: 2 999 dollár, amit arany szolgáltatásnak hívnak.

100 emberre egy éves használati díj ára: 1 999 dollár, amit ezüst szolgáltatásnak hívnak.

Red Hat Application Stack

Az előbbieket Linux megvalósítása és ezt a megvalósítást nevezik Red Hat Application Stack-nek, ami a következő technológiák ötvözet:

- Apache server
- JBoss Application Server
- JBoss Hibernate
- MySQL és PostgreSQL
- Red Hat Enterprise Linux

Lehetőség nyílik arra, hogy a Red Hat Enterprise Linux-ot és a adatbázis szerveret külön vásároljuk meg.

Red Hat Application Stack ára az első évben 1 000 felhasználóra: 8 500 dollár, ami a prémium csomagot tükrözi.

Red Hat Application Stack ára szintén az első évben 100 felhasználóra: 2 000 dollár, ami az alap csomagot tükrözi.

Alfresco

Nyílt forráskódú programozást használ, amivel csökkenteni tudja a kiadásokat.

Ilyen a:

- csökkentett fejlesztési idő: sok fejlesztővel rendelkezik, akik igazi „öreg rókának” számítanak, erős csoport áll mögötte. Ezen felül további nyílt forráskódú projektekkal működik együtt, hogy csökkentse a fejlesztési kiadásokat. Ez azt eredményezi, hogy fejlettebb teszteléseket végezhetnek, ezáltal minőségileg, funkcionalításban gyorsabban fejlődnek. A nyílt módszernek hála, további nyílt forráskódú elemek implementálását is támogatja.
- Ingyenes próbaverziók: ingyenes termékek használatával megengedjük leendő ügyfeleink számára, hogy akár valós körülmények között teszteljék a programot.

Alfresco stabilitásának bizonyítására egy független cég teljesítménymérést végzett el rajta. Ennek eredményeként 2008-ban bejelentették, hogy túllépték a 100 milliós dokumentum szintet. Erre egy nyolc utas Unisys ES7000 gépet használtak szerverként, amiben egy dual core Xeon processzor volt található, egy MySQL adatbázis, és a Red Hat Enterprise OS. Ez bizonyította Alfresco az gyorsaságát. A mért adatok:

- Állandó 140 objektum/mp
- Lineáris skálázhatóság nagy adattárhely keresésekkor is.
- Másodpercen aluli hozzáférési idők a standard felhasználóktól ftp és http protokollokon is
- Tranzakciók folyamatosak voltak, biztonságosak még 100 milliós objektum felöltés alatt is

Összefoglalás

A nyílt forráskódú programozás mindenhol elterjedt Manapság már nélkülük lehetetlen elképzelni az egyes megoldásokat. Sok újítást hoztak, amit az XML népszerűségének és a sikeres programozási nyelvek adaptálásával valósulhatott meg.

Ez a fajta kombináció olyan jól sikerült, hogy az új cégek legnagyobb százaléka ezt használja fel, alapként a saját munkájához. Így tett az Alfresco és számos egyéb rendszer is, és észrevehetően, ez nagyon is jó megoldás lett. Ennek hála nagyon gyorsan az élvonalba tudott kerülni az ECM szolgáltatóknál, a konkurens céghez mért árakról nem is beszélve. Az Alfresco a szabványok által előírt úton halad, ezt a legtöbb helyen adaptálják, és már nem kell sokat várni, ahhoz hogy, az egyik legnagyobb ECM szolgáltató legyen.

1000 User Enterprise Configuration						
Vendor	Configuration	ECM Cost	Combined Stack Cost	Total Year One Cost	Alfresco Year One Saving	Percentage Year One Saving
Documentum	Oracle+Windows (Assume Application Server Included)	\$863,937	\$61,151	\$925,088	\$887,790	96%
OpenText	Oracle+Windows	\$637,304	\$61,151	\$698,455	\$661,157	95%
SharePoint	2 SQL Server Standard Edition + 3 Windows Servers	\$318,738	\$20,411	\$339,149	\$301,851	89%
Alfresco	Red Hat Application Stack Standard Subscription	\$33,500	\$5,499	\$38,999		
Alfresco	Red Hat Enterprise Linux Standard + MySQL Gold	\$33,500	\$3,798	\$37,298		
1000 User Clustered High Availability Enterprise Configuration						
Vendor	Configuration	ECM Cost	Combined Stack Cost	Total Year One Cost	Alfresco Year One Saving	Percentage Year One Saving
Documentum	Oracle+Windows (Assume Application Server Included)	\$863,937.00	\$61,151	\$925,088	\$875,040.00	95%
OpenText	Oracle+Windows	\$637,304.00	\$61,151	\$698,455	\$648,407.00	93%
SharePoint	2 SQL Server Standard Edition + 3 Windows Servers	\$318,738.00	\$20,411	\$339,149	\$289,101.00	85%
Alfresco	Red Hat Application Stack Standard Subscription	\$46,250.00	\$5,499	\$51,749		
Alfresco	Red Hat Enterprise Linux Standard + MySQL Gold	\$46,250.00	\$3,798	\$50,048		
100 User Basic Configuration						
Vendor	Configuration	ECM Cost	Combined Stack Cost	Total Year One Cost	Alfresco Year One Saving	Percentage Year One Saving
Documentum	1 SQL Server Standard Edition + 2 Windows Servers	\$129,078	\$10,781	\$139,859	\$119,360	85%
OpenText	1 SQL Server Standard Edition + 2 Windows Servers	\$196,754	\$10,781	\$207,535	\$187,036	90%
SharePoint	1 SQL Server Standard Edition + 2 Windows Servers	\$24,669	\$10,781	\$35,450	\$14,951	42%
Alfresco	Red Hat Application Stack Basic Subscription	\$18,500	\$1,999	\$20,499		
Alfresco	Red Hat Enterprise Linux Basic + MySQL Silver	\$18,500	\$2,348	\$20,848		

10

3. kép: ECM rendszerek összetett árazása három aspektus alapján

Nyíltság fogalma

Olyan sokat hallhattunk már nyílt keresésről, nyílt programozásról, nyílt forrásokról, de még nem is tudjuk, hogy valójában mit is takar. Kulcsfontosságú hogy megértsük, és tisztán lássuk a fogalom jelentését, és azt hogy mi is áll mögötte valójában. Egy kis szemantikai fejtegetéssel hadd bővítsem munkámat.

¹⁰ 3.kép: ECM TCO 15. oldal 2009.04.02.

„Nyílt, vagyis nem szabad!” A két kifejezést nem szabad összekeverni, habár manapság már nagyon egybecseng „hála” egyes törekvéseknek. Itt, ebben a kontextusban a szabadság értelme annyit tesz, hogy szabadon használatom, fejleszthetem és akár terjeszthetem is a programokat. Ilyen értelemben már szinte megközelíti az igazi szabadság fogalmát, melyről a legnagyobb művészek ábrándoznak műveikben. E gondolat köré hozták létre az általános nyilvános licenst, (General Public Licence, GPL¹¹), amely a licenz hatálya alá tartozó programok felhasználói jogait méltatott védeni. A 80-as évek közepétől egy új mozgalom volt kialakulóban, akik céljuknak tekintették egy olyan környezet(ek) kiépítését, ami teljesen megfelel a GPL-nek. Sajnos idővel kiváltak emberek, akik megéreztek az üzleti esélyeit ennek a rendszernek, és forgalomba hozták a saját termékeiket. Ők kezdték el a nyílt forrást, mint a szabad forrás alternatíváját hirdetni, erősen gazdasági megfontolásból. A nyílt forrás kifejezés hamar a megbízhatóság és az erő záloga lett. Most már láthatjuk, hogy milyen háttér áll az egyes kifejezések mögött. A szabad program ingyenes és a felhasználó teljes mértékben rendelkezhet felette, míg a nyílt forrás mai szemmel nézve egy reklám stratégia kifejezetten jó terméke. Mindkét kifejezés ugyan arra a mögöttes technológiára utal, csak két különböző aspektusból.

A nyílt forrású program (open source software) kifejezés tartalmilag azt sugallhatja, hogy a program kódolása látható. Sokan azt hitték, - bevallom eddig én is - , hogy ez ennyiből áll. De kutatásaim során, rádöbentem, hogy ennél azért jóval többről van szó, hisz az ami igazán a nyílt forrás mögött áll, az, az, hogy bárki bár mikor hozzáférhet, változtathat a programon, és ezt mind ingyen megtehetik. Ez új távlatokat nyit a fejlesztésben, hisz bárki a szerkesztőjévé válhat, és sokkal egyedibb, felhasználóbarátabb programok születhetnek, aminek végső soron mi, a felhasználók örülünk.

Ennyit a kifejezések szemantikai hordozásáról és különbségeikről, rátérnék a megvalósításra. Szerencsére, itt nem fog kelleni sokat írni erről, ugyanis tényleg azt jelenti a nyílt forráskód, hogy rendelkezem a program futásához kellő összes kóddal, olyan formában, amit akár egy egyszerű jegyzetömbbel is megnyithatunk és változásokat eszközölhetünk rajta. Ilyen programozási nyelvek, melyet az Alfresco

¹¹ <http://www.gnu.org/licenses/gpl.html> 2009.05.12. itt megtalálható a magyar fordításhoz a link

felhasznál a következők: ASP, JSP, JavaScript és AJAX¹². Ezek mindegyike megfelel annak a követelménynek, hogy szabadon hozzáférhető és látható. De egy kivétel, egy fontos kivétel még hiányzik. Ez a változtatás és terjesztés kivétele. Ez a GPL-nek hála érvényesül, ugyanis a legtöbb Alfresco program ezt már tartalmazza. Kira-gadtam az Alfresco login.asp program e szeletét. Ezt azért is tehetem meg, ugyanis a licenz ezt is megengedi.

```

login - Notepad
File Edit Format View Help
k%--
* Copyright (C) 2005-2007 Alfresco Software Limited.

* This program is free software; you can redistribute it and/or
* modify it under the terms of the GNU General Public License
* as published by the Free Software Foundation; either version 2
* of the License, or (at your option) any later version.

* This program is distributed in the hope that it will be useful,
* but WITHOUT ANY WARRANTY; without even the implied warranty of
* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
* GNU General Public License for more details.

* You should have received a copy of the GNU General Public License
* along with this program; if not, write to the Free Software
* Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301, USA.

* As a special exception to the terms and conditions of version 2.0 of
* the GPL, you may redistribute this Program in connection with Free/Libre
* and open source software ("FLOSS") applications as described in Alfresco's
* FLOSS exception. You should have recieved a copy of the text describing
* the FLOSS exception, and it is also available here:
* http://www.alfresco.com/legal/licensing"
--%>
<%%@ taglib uri="http://java.sun.com/jsf/html" prefix="h" %>
<%%@ taglib uri="http://java.sun.com/jsf/core" prefix="f" %>
<%%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%%@ taglib uri="/WEB-INF/alfresco.tld" prefix="a" %>
<%%@ taglib uri="/WEB-INF/repo.tld" prefix="r" %>

<%%@ page import="org.alfresco.web.app.servlet.BaseServlet" %>
<%%@ page import="org.alfresco.web.app.servlet.AuthenticationHelper" %>
<%%@ page import="org.alfresco.web.ui.common.PanelGenerator" %>
<%%@ page import="org.alfresco.web.app.Application" %>
<%%@ page import="javax.faces.context.FacesContext" %>
<%%@ page import="javax.servlet.http.Cookie" %>
<%%@ page import="java.util.Locale" %>

```

4. kép: login.jsp licenz részlete, ami a C:\Alfresco\tomcat\webapps\alfresco\jsp könyvtárban található

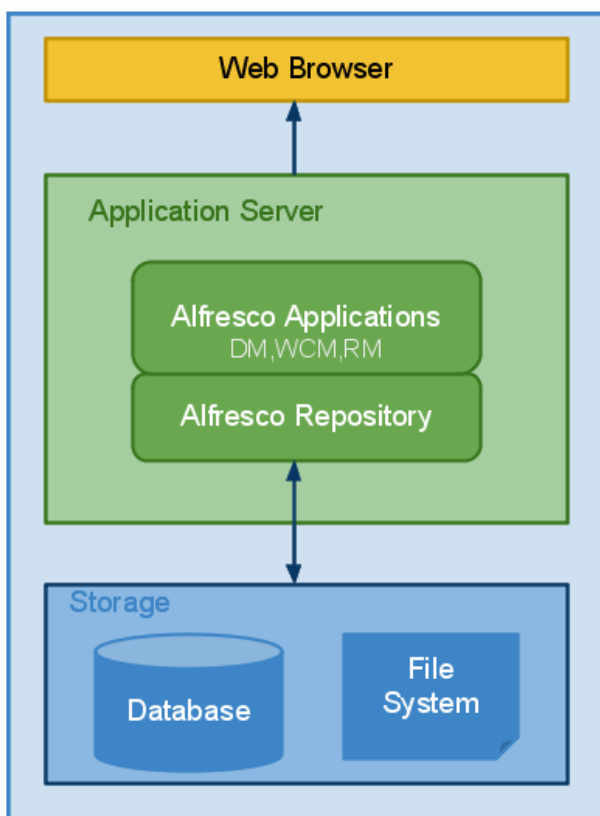
Tovább haladva, közelítsük meg a program felépítését, a használt technológiákat és a támogatott interfészeket (interface). Az első és legfontosabb az XML, ami a program beállításait, lehetőségeit tartalmazza. Szinte minden adat, ami nem programrészlet vagy egyéb cég formátuma, az XML dokumentumban definiálva van. További technológia: Hibernate, JLAN, OpenOffice, POI File Format

¹² <http://java.sun.com/developer/technicalArticles/J2EE/AJAX/>

Conversion, Spring stb. Támogatott interfészek: CIFS, FTP, JSR-127, JSR-168, REST, WebDav.

Sajnos az itt felsorolt elemek közül csak az XML-t és az OpenOffice-t ismerem. Pontosabban a felsoroltak közül e kettő bemutatására teszek kísérletet, a többi sajnos csak említés szintjén tudom leírni. Érthető módon tanulmányaim nem tértek ki ilyen vagy esetleg hasonló témákra, így meg sem kísérem az elemzésüket. Mindenesetre olvastam róluk és egy-két mondat erejéig említést teszek róluk. Ahhoz, hogy azért valamiféle rálátást adjak a technológiák mivoltára és szükségességére, ezért egy-két ábra magyarázatával szeretném valamiféleképpen e hiányomat kompenzálni. Szerintem a legjobb módja az lesz, ha az Alfresco felépítését mutatom be.

Alfresco felépítése



5. kép: Alfresco egyszerű felépítése

Alfresco telepítése után, egy 5. kép felépítésű rendszert kapunk. Ahogy azt jól látni a webszerverünk áll középen, és rajta keresztül történik a felhasználó interakciói a rendszerrel. Az adatokat vagy egy adatbázis szerver vagy egy fájl rendszer

kezeli. Ilyen fájl rendszert XML-ből is kialakíthatunk. Az egész lényege, hogy JSP vagy lehet szervlet is, a tomcat webszerver felfutása után betöltődik, és egy kapcsolatot létesít az adatbázissal. Majd a felhasználó a böngészőjét használva kommunikálhat a számítógéppel. Ahhoz, hogy a lehető legjobb és legkellemesebb szolgáltatásokat nyújthassa, ahhoz van szüksége a különböző technológiákra. Meg kell említeni, hogy az Alfresco-nak minden része: vagy egy komponens (ha komponens, akkor valamilyen technológia) vagy egy szolgáltatás (ha szolgáltatás, akkor az JSP). Ezt a felépítést a Spring Framework¹³ nevezetű de facto szabvány teszi lehetővé, ami az alapja az Alfresco architektúrájának. (6. kép).

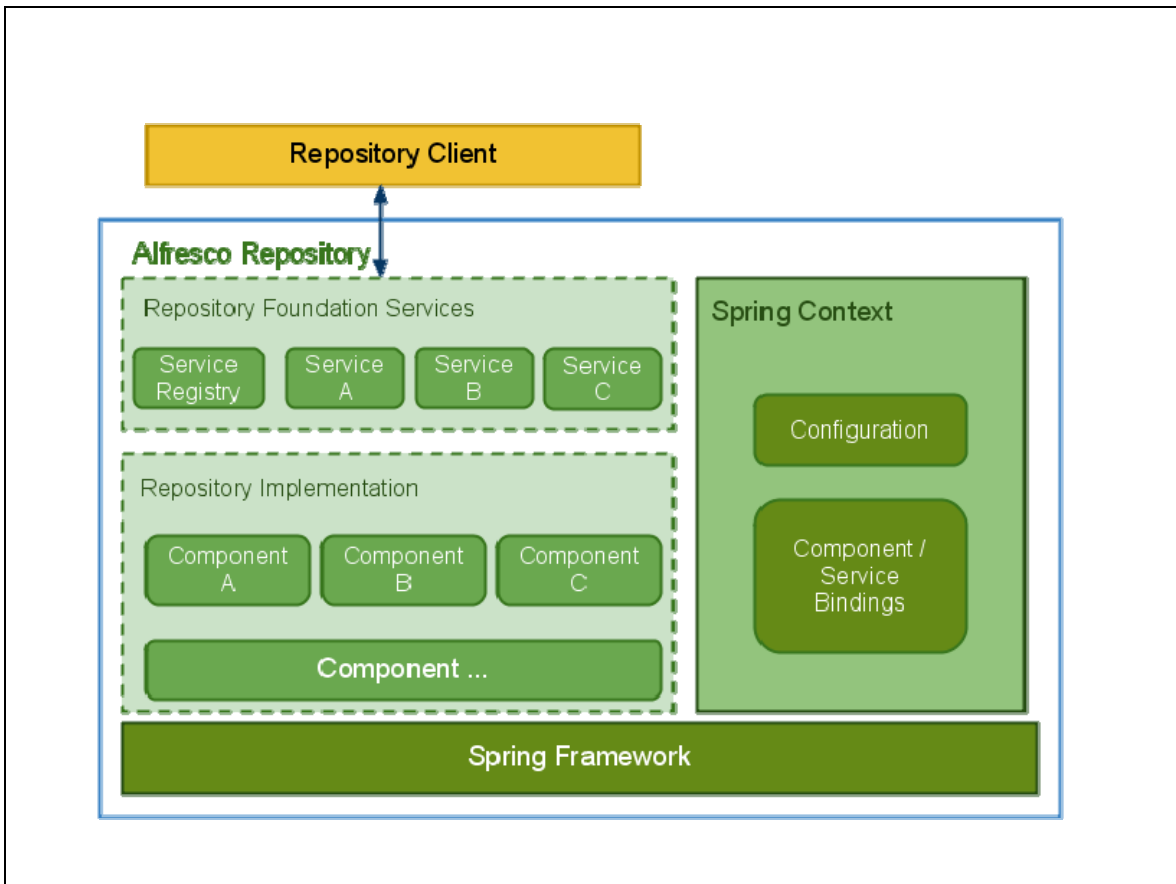
Az Alfresco adattárhely felelős a tartalom kezeléséért, azaz visszakeresésért és tárolásért. Ennek megvalósítására alapvető szolgáltatások (szolgáltatás bejegyzés, csomópont-, tartalom-, keresési- és egyéb szolgáltatások) kellene (7. kép). Az alapvető szolgáltatások egy alkalmazásprogramozási felületet (Application Programming Interface, API) képeznek, mely előnye, hogy egy programot használhatunk úgy is, hogy nem ismerjük a belső működését. Azaz erre implementálva ez azt jelenti, hogy mi beprogramozhatunk, megírhatunk egy FTP szerveret anélkül, hogy tudnánk, hogy annak működése, hogy valósul meg az adatraktár protokoll szerverében. Még egyszerűbben leírva: a Google-ben keresek, de nem tudom, hogy milyen keresési és rangsorolási algoritmussal dolgozik. Az alapvető szolgáltatásokat az egyes technológiák teszik (CIFS¹⁴, FTP, WebDAV¹⁵) lehetővé. Ezeknek a technológiáknak a lehetőségeit használják majd fel az egyes kommunikációs protokollok (FTP kliens, WebDAV kliens) kielégítésére (7. kép). Ahogy látjuk, itt már olyan protokollokról van szó, mint az FTP¹⁶ (File Transfer Protocol), amelyeket már közvetlenül elérhetünk. Mint minden mást, ezt is a Spring Framework irányítja.

¹³ http://en.wikipedia.org/wiki/Spring_Framework 2009.05.13.

¹⁴ <http://www.softwareonline.hu/Article/View.aspx?id=2989> 2009.05.13.

¹⁵ <http://en.wikipedia.org/wiki/WebDAV> 2009.05.13.

¹⁶ http://hu.wikipedia.org/wiki/File_Transfer_Protocol 2009.05.13.

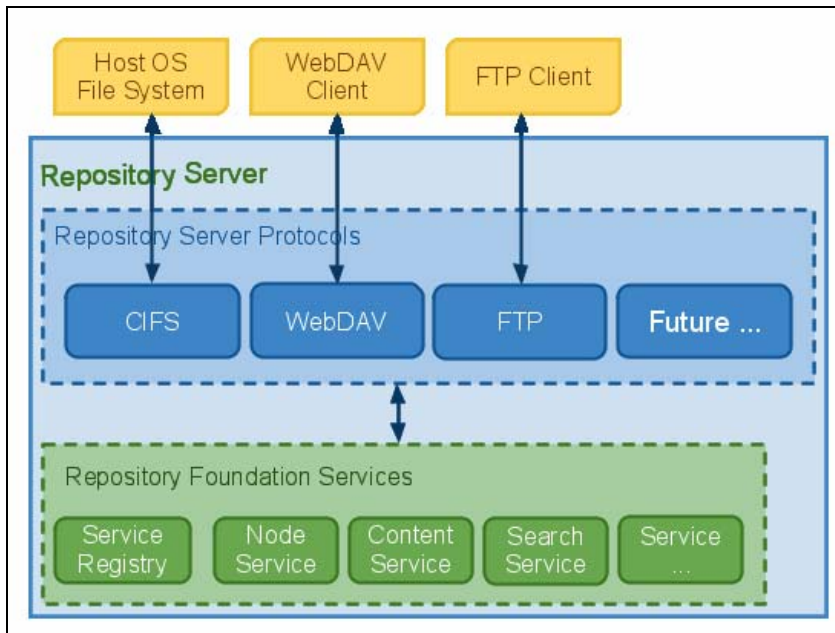


6. kép: Alfresco Spring Framework architektúrája, és elemek

Alfresco további két API-t bocsát rendelkezésünkre. Az egyik a JCR (Java Content Repository) ami igazából a JSR-170¹⁷ -es szabvány, és a Web Services API. A Web Services API-nak hála, szinte bármilyen programozási nyelvvel hozzáférhetünk. Ilyen lehet az ASP, PHP, .NET és természetesen Java.

Ehhez a szekcióhoz leírtam mindent, amit tudtam, remélem – legalább - a képek nyújtanak majd valamiféle támpontot. Ezek után áttérnék az XML tárgyalására, ami egy következő fundamentális alapja a rendszernek.

¹⁷ <http://jcp.org/aboutJava/communityprocess/final/jsr170/index.html>



7. kép: szolgáltatások és kommunikációs protokollok, majd azok végső szolgáltatásai

XML

Bővíthető jelölőnyelv¹⁸ (Extensible Markup Language) a Word Wide Web Consortium (W3C) terméke, mely egy olyan szintakszist ad meg, amelyet betartva különböző jelölő nyelvek (mint például az XHTML) hozhatóak létre. Éppen ezért szokták meta-nyelvnek, nyelvleíró nyelvnek is nevezni. Őse az SGML (Standard Generalized Markup Language¹⁹) származtatott jelölőnyelv, amelyet 1996-ban kezdték el fejleszteni. A W3C XML 1.0 ajánlása 1998.02.10-én született meg. Bővíthetősége azt tükrözi, hogy saját elemeket lehet benne létrehozni, vagy (Document Type Definition, DTD), vagy XML séma létrehozásával. A HTML-be (HyperText Markup Language²⁰) folyamatosan bekerülő újabb és újabb, formai jelölésekre szolgáló elemek lassan, tartalmi szempontból teljesen feldolgozhatatlanná tették az interneten elérhető dokumentumokat. Ezen problémák kiküszöbölésére alkották meg az XML-t. Követelmény volt, hogy az XML dokumentum jól formázott, embe-ri olvasásra is alkalmas legyen. Egy XML dokumentumnak rendelkezni kell (kellene) DTD²¹ vagy XML sémával (XML Schema, XSD²²). XML dokumentum ele-

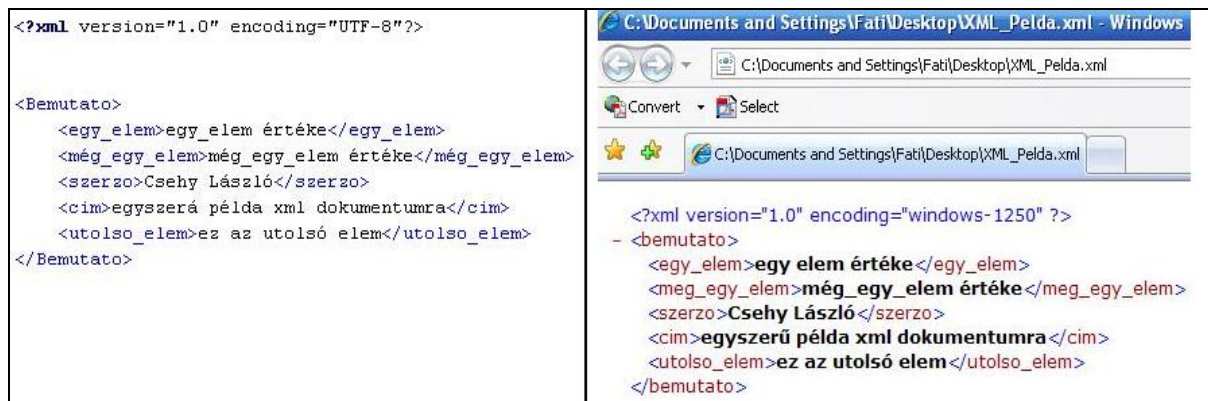
¹⁸ <http://www.w3.org/XML/Core/#Publications> 2009.05.02.

¹⁹ <http://www.w3c.org/MarkUp/SGML/> 2009.05.02.

²⁰ <http://www.w3.org/TR/REC-html40/> 2009.05.10.

²¹ <http://hu.wikipedia.org/wiki/Jelolonyelv> 2009.05.10.

meből áll, amelyeknek neveket adunk A nevekre, vagyis XML-ben címkékre (tag) úgy lehetne tekinteni mint egy szótár szavaira. Az egyes címkék egymáshoz való kapcsolatai tartalmi szabályokkal rögzíthetők (nyelvtan). Az XML specifikáció megad egy szintaxist az XML dokumentumokra, az elemei jelölésére és a szabályok leírására (DTD vagy XSD). A megadott szintaktikai szabályok betartásával bárki saját nyelvet (dokumentumot) készíthet, s azt a megfelelő XML validator (ellenőrző) eszközzel leellenőrizheti. Az XML kiterjeszhetősége lehetőséget ad a dokumentumok (adatok) tartalmi és nem formai szempontok alapján történő leírására is. XML-ben deklarálhatunk egy `<laci></laci>` vagy egy `<LADISLAV></LADISLAV>` elemet is.



8. kép: bal oldalt egy egyszerű XML leírás, míg jobb oldalt a megfelelője IE böngészőben

Emiatt és ennek okáért rengeteg új eszköz, lehetőség látott rövid idő alatt napvilágot, és a világ „nyíltá” vált.

Formai elvárások

Az XML dokumentumnak formailag két szempontnak kell, hogy megfeleljen, egyik hiánya máris használhatatlanná teszi:

- Jól formáltság követelménye: Olyan XML dokumentum , amely megfelelően beágyazott címkéket tartalmaz, és minden objektum előtt és után szerepel a nyitó és záró jelölőelem. Ezenfelül a dokumentumnak,

²² <http://www.w3.org/XML/Schema>

teljesnek is kell lennie (nem hivatkozhat a dokumentumon kívüli entitásokra).

- Érvényesség követelménye: XML dokumentumnak jól formázottnak kell lennie és megfelelnie az őt meghatározó DTD vagy XSD fájlra. Ha ez teljesül akkor válik érvényessé a fájl.

A fentiekből kiderült, hogy az XML és a HTML nem egy szinten van. Az XML egy szintaktikai szabályrendszer, amellyel leíró nyelveket definiálhatunk különböző alkalmazásokhoz (például egy olyat, mint a HTML). A HTML ezzel szemben egy SGML alkalmazás, egy nyelv, amelyet SGML segítségével definiáltak, és azt a célt szolgálja, hogy egy platform- és eszköz független dokumentum formátumot adjon az Internetnek. Az XML kifejlesztését az SGML bonyolultsága tette szükségessé.

Az XML elsősorban az internetre készült, így alkalmazható különböző dokumentumok „webes” publikálására, feldolgozására. Formátuma lehetővé teszi automatikus generálhatóságát különböző más formátumokra, mint például HTML, PDF (Portable Document Format)

Felhasználási területei az XML-nek:

- Szövegfeldolgozás: Az SGML „szakterülete”, de természetesen az XML is felhasználható különböző szövegek elektronikus leírására, sőt szerintem mára a legelterjedtebbé vált. Megfelelő tartalmi jelölésekkel a megjelenítésen túlmenő, mélyebb gépi feldolgozásra is alkalmas dokumentumokat lehet elő állítani. Például: DocBook²³
- „Nagy” mennyiségű adat tárolására az XML már nem igazán ajánlatos, erre a feladatra lehetőség szerint adatbázis-kezelő szoftverek alkalmasak, de evvel szemben az XML adatbázisok jobb és erőteljesebb megoldást biztosítanak a tárolt adtok manipulációjára és tárolására, mint a relációs adatbázis-kezelők, olyan alkalmazások részére melyek XML adatokkal dolgoznak

²³ <http://www.docbook.org/>

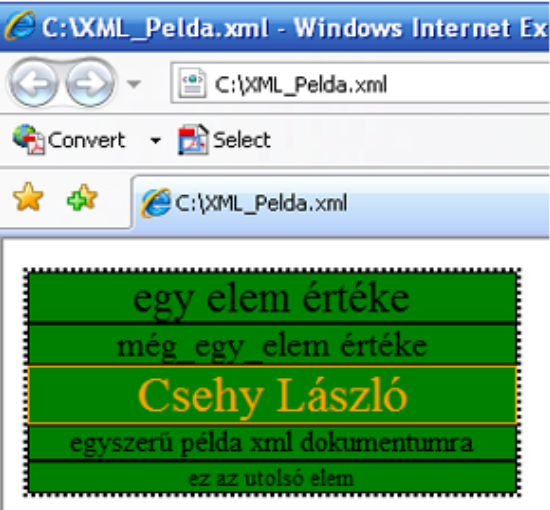
- A függetlensége, elterjedése és a lehetőségei szinte elvárják, hogy mintegy közös nyelvként funkcionáljon. Mintha egy köztes réteg lenne az egyes már használatban lévő formátumok cseréjére.
- Egy protokoll szintaxisa is lehet XML. Ilyen például a XML-alapú protokoll a SOAP²⁴ (Simple Object Access Protocol, egyszerű elem hozzáférési protokoll), amely web-service alkalmazások közti kommunikációra használatos
- Strukturált dokumentumok (referátumok, kézikönyvek) jelölésére, összetett adatok szoftver alkalmazások közti átvitelre, elsősorban az Interneten keresztül. XML adat önleíró jellege alapvető fontossággal bír
- Konfigurációs fájlok számára ideális adatformátum.
- Multimédia adatformátum, segítségével multimédiás prezentációkat készíthetünk

XML dokumentum ahogy a 8. képen láthattuk megjelenésében kicsit szegényes. Ez a látszólagos hiányosság éppen az SGML filozófiájának lényegéből, a tartalom és a forma különválasztásából ered. A nyelv nem tudja megfelelő mértékben feltárni a dokumentumok megjelenítését, így a web designerek esztétikai törekvései csak szűk keretek között valósulhatnak meg. Ha az adatok csak gépeknek íródtak volna, akkor ez a formai megvalósítás még eltúlzottnak is vélhetnénk, de nem mindegyiket, vagyis sokukat pont arra terveztek, hogy szolgáltatassák a tartalmukat. Így amit bejön az emberi faktor, máris egy sokkal kellemesebb küllemmel kell felruháznunk az XML dokumentumunkat. Erre a célra fejlesztették ki az egymásba ágyazott stíluslapokat²⁵ (Cascading Style Sheet, CSS) melyekkel már igazán szép külalakot biztosíthatunk dokumentumainknak. Oldalak megjelenését befolyásoló egyszerű jelölőnyelv, mely segítségével meghatározhatjuk az egyes címkék, elemek, attribútumok megjelenési helyét az oldalon. Ezt tükrözi a 9. kép is. A CSS ajánlás leírása 1996. december 17. jelent meg a W3C égisze alatt. Továbbá már az

²⁴ Üzenetküldésre használt XML alapú formátum, amelyet különböző protokollok segítségével továbbíthatjuk, a legelterjedtebb a http protokoll. W3c tartja karban: <http://www.w3.org/TR/soap/>

²⁵ <http://www.w3.org/Style/CSS/>

XML szintaktikára építve létrehozta egy egyszerű szedési nyelvet, amely a CSS-hez hasonló formázási lehetőségeket biztosít. Szintén a W3C terméke, melynek az 1.0 publikációja 2001-ben jelent meg. Azóta 2008-ban megjelent a 2.0 tervezete is. Az XSL-t bemutató példa a 10. kép.

<pre><?xml version="1.0" encoding="windows-1250"?> <?xml-stylesheet type="text/css" href="stiluslapneve.css" ?> <bemutato> <egy_elem>egy elem értéke</egy_elem> <meg_egy_elem>még egy elem értéke</meg_egy_elem> <szerzo>Csehy László</szerzo> <cim>egyszerű példa xml dokumentumra</cim> <utolso_elem>ez az utolsó elem</utolso_elem> </bemutato></pre>	<pre>bemutato { width: 250px; display: block; text-align: center; background: green; border: dotted 2px; } egy_elem { display: block; border: solid 1px; font-size: 22px; } meg_egy_elem { display: block; text-color: orange; border: solid 1px; font-size: 18px; } szerzo { display: block; color: orange; cursor: pointer; border: solid 1px; font-size: 25px; } cim { display: block; border: solid 1px; font-size: 15px; } utolso_elem { display: block; border: solid 1px; font-size: 12px; }</pre>
	

9. kép: egy egyszerű külső css formázás hozzáadása az XML dokumentumhoz

Értelmezők / validator

Ilyen és a jóval bonyolultabb kódok írása esetén elég egy kis figyelmetlenség és máris egyszerűbb esetben rosszul formázott, rosszabb esetben pedig érvénytelen dokumentumot kapunk. Szerencsére a piacon már sok igazán jó teljes körű XML feldolgozó található, ilyen a:

- Altova XMLSpy²⁶
- ElCel Technology XML Validator²⁷

²⁶ <http://www.altova.com/products/xmlspy/xmlspy.html>

²⁷ <http://www.elcel.com/products/xmltools/xmlvalid.html>

- Liquid XML Studio²⁸

Léteznek internetes ellenőrzők is, legmegbízhatóbbak:

- <http://www.w3.org/2001/03/webdata/xsv> 2009.05.11.
- <http://www.validome.org/xml/> 2009.05.11.
- http://www.w3schools.com/XML/xml_validator.asp 2009.05.11.

Példák

Nézzünk egy másik XML példát és a DTD által meghatározott elemeit:

```
<?xml version="1.0" encoding="windows-1250" standalone="yes"?>
  <?xml-stylesheet type="text/xsl" href="ember2.xsl"?>
  <!DOCTYPE ember [
    <!ELEMENT ember (szemely)+>
    <!ELEMENT szemely ( neve, kora, (lakhely | tartozkodasi_hely)*, sz_ig, magassag, suly)>
    <!ELEMENT neve (#PCDATA)>
    <!ELEMENT kora (#PCDATA)>
    <!ELEMENT lakhely (#PCDATA)>
    <!ELEMENT tartozkodasi_hely (#PCDATA)>
    <!ELEMENT sz_ig (#PCDATA)>
    <!ELEMENT magassag (#PCDATA)>
    <!ELEMENT suly (#PCDATA)>

    <!ATTLIST lakhely ir_sz CDATA #REQUIRED>
    <!ATTLIST tartozkodasi_hely ir_sz CDATA #REQUIRED>
    <!ATTLIST szemely neme (f|n) "f">
    <!ATTLIST kora publikus (i|n) "i">
    <!ATTLIST sz_ig publikus (i|n) "i">
    <!ATTLIST magassag me (cm|m|láb|hüvelyk) "cm">
    <!ATTLIST suly me (g|dkg|kg|q) "kg">
  ]>

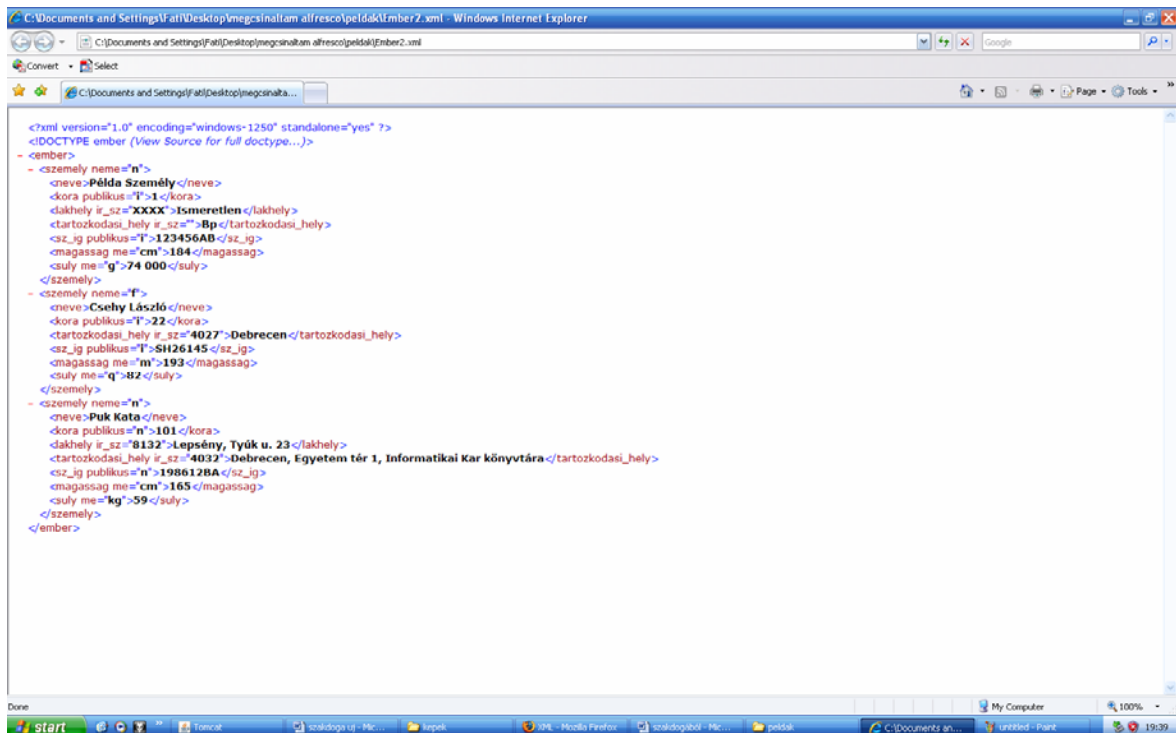
  <ember>
    <szemely neme="n">
      <neve>Példa Személy</neve>
      <kora>1</kora>
      <lakhely ir_sz="XXXX">Ismeretlen</lakhely>
```

²⁸ <http://www.liqviudtechnologies.com/XmlStudio/XmlStudio.aspx>

```

<tartozkodasi_hely ir_sz="">Bp</tartozkodasi_hely>
<sz_ig>123456AB</sz_ig>
<magassag>184</magassag>
<suly me="g">74 000</suly>
</szemely>
<szemely>
  <neve>Csehy László</neve>
  <kora>22</kora>
  <tartozkodasi_hely ir_sz="4027">Debrecen</tartozkodasi_hely>
  <sz_ig>SH26145</sz_ig>
  <magassag me="m">193</magassag>
  <suly me="q">82</suly>
</szemely>
<szemely neme="n">
  <neve>Puk Kata</neve>
  <kora publikus="n">101</kora>
  <lakhely ir_sz="8132">Lepsény, Tyúk u. 23</lakhely>
  <tartozkodasi_hely ir_sz="4032">Debrecen, Egyetem tér 1, Informatikai
  Kar könyvtára</tartozkodasi_hely>
  <sz_ig publikus="n">198612BA</sz_ig>
  <magassag>165</magassag>
  <suly>59</suly>
</szemely>
</ember>

```



10. kép: a példa „nyers” XML megjelenítése


```

<tr>
  <td style="border:none; padding-left:5px; padding-right:10px">
    <b>Személyigazolvány száma:</b>
  </td>
  <td style="border:none">
    <xsl:choose>
      <xsl:when test="sz_ig/@publikus='i'">
        <xsl:value-of select="sz_ig" />
      </xsl:when>
      <xsl:otherwise>
        <i>--- nem publikus ---</i>
      </xsl:otherwise>
    </xsl:choose>
  </td>
</tr>
<tr>
  <td style="border:none; padding-left:5px"><b>Magassága:</b></td>
  <td style="border:none">
    <xsl:value-of select="magassag" />&#160;
    <xsl:value-of select="magassag/@me" />
  </td>
</tr>
<tr>
  <td style="border:none; padding-left:5px">
    <b>Súlya:</b>
  </td>
  <td style="border:none">
    <xsl:value-of select="suly" />&#160;
    <xsl:value-of select="suly/@me" />
  </td></tr></table><br/></xsl:for-each>
</BODY></HTML></xsl:template></xsl:stylesheet>

```



11. kép: XSL stíluslap használata XML dokumentumon

Az XML ismereteinknek most fogjuk hasznát venni, ugyanis a legtöbb nyílt forráskód erre épít. Úgy is mondhatnám, hogy a program futásához és beállításához szükséges adatokat XML formátumú fájlok tömegeiben tárolják. Ezekből állandóan a különböző műveleteknek megfelelően kikérhetők az adatok. Az XML stabilitásnak, egyértelműségének, használhatóságának hála szinte minden adatot el tudnak tárolni. Ha rendelkezünk egy nyílt programmal, akkor annak „home” könyvtárában biztos találunk az Alfresco XML fájljához hasonló dokumentumokat.

```
</mimetype>
- <mimetype mimetype="image/ief" display="IEF Image">
  <extension>ief</extension>
</mimetype>
- <mimetype mimetype="image/bmp" display="Bitmap Image">
  <extension>bmp</extension>
</mimetype>
- <mimetype mimetype="image/jpeg" display="JPEG Image">
  <extension default="true">jpg</extension>
  <extension>jpeg</extension>
  <extension>jpe</extension>
</mimetype>
- <mimetype mimetype="image/svg" display="Scalable Vector Graphics Image">
  <extension>svg</extension>
</mimetype>
- <mimetype mimetype="application/x-javascript" text="true" display="Java Script">
  <extension>js</extension>
</mimetype>
- <mimetype mimetype="application/x-latex" text="true" display="Latex">
  <extension>latex</extension>
</mimetype>
- <mimetype mimetype="application/x-troff-man" display="Man Page">
  <extension>man</extension>
</mimetype>
- <mimetype mimetype="application/x-troff-me" display="">
  <extension>me</extension>
</mimetype>
- <mimetype mimetype="application/x-troff-mes" display="">
  <extension>ms</extension>
</mimetype>
- <mimetype mimetype="application/x-mif" display="">
  <extension>mif</extension>
</mimetype>
- <mimetype mimetype="video/mpeg" display="MPEG Video">
  <extension default="true">mpg</extension>
  <extension>mpeg</extension>
  <extension>mpe</extension>
```

12. kép: Alfresco által használt MIME típusok listája

Miután a rendszer felépítésének leírtam az alapvető részeit és annak funkcióit, munkám hátralévő részét a program bemutatásának szentelem.

Tovább haladva, meg kell említenem a **nyílt dokumentum formátumot** is. Ugyanis a dokumentum és tartalom lételeme a szervezeteknek, kormányoknak és lakosoknak. Arra használják őket, hogy tudást, fontos információt tároljanak, tevé-

kenységet hangoljanak, eredményeket hasonlítsanak össze, és az ebből nyert eredményeket a külvilág felé publikálják. Az a legfontosabb, hogy ezeket a belső és külső dokumentumokat cserélni, szolgáltatni lehessen a szervezeteken belül és ne okozzon fennakadást az, ha valaki más szoftvert, technológiát, árlistát használ, ez ne egy külső cég döntésétől függjön. Ezen felül a lakosok képesek legyenek a tartalmat elérni anélkül, hogy bármiféle megkötésük legyen a technológiából vagy formátumból adódóan. Ehhez szükségünk van ahhoz, hogy tudjuk:

- hogy lehet könnyen megosztani a tartalmat: lakosok, szervezetek és kormányok között
- hogy lehet majd a dokumentumot nézni, használni, módosítani a jövőben
- milyen jogai lesznek a felhasználónak a jövőben, azzal kapcsolatban hogy láthassa, használja és módosítsa a dokumentumot a jövőben ha már rég lejárt a jogviszonya az adott szolgáltatóval

Erre megoldása az Open Document Format²⁹ (nyílt dokumentum formátum, röviden ODF), ami rövid és hosszú távú követelményekre jelent megoldást. Az ODF egy XML alapú fájl formátum, ami irodai formátumok használatakor alkalmas, amit arra terveztek, hogy a szöveget, táblázatkezelőt, diagramokat, grafikus dokumentumokat és az ezek által megkövetelt sajátosságokat kezelje. A szabvány OASIS Open Document Technical Committee által van felügyelve, melyet a nyilvános XML szabványok fejlesztésére hozták létre. Ez egy OASIS által elfogadott szabvány, melyet az ISO, IEC, JTC1-hez is benyújtottak, hogy a jövőben egy elfogadott de jure szabvány legyen. Teljesen ingyenes és bárki használhatja, ezért és ezekért a tulajdonságai miatt lett elfogadott a de facto választások sok helyen. Az ODF célja, hogy egy alkalmazás független formátumot nyújtson, ami szolgáltató független is egyben. Sok irodai szolgáltató használja saját munkájában az OpenOffice.org XML formátumot. Az Alfresco továbbá rendelkezik egy intelligens virtuális fájl rendszerrel (Intelligent Virtual File System), ami egy egyszerű megosztott meghatót nyújt az adatraktárban egy felhasználói felületen keresztül. Ez lehetővé teszi, hogy a saját szerkesztői eszközeit használja (Microsoftot vagy egyebet) egyszerűen csak „men-

²⁹ http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office 2009.05.12

tés” –t vagy „drug-and-drop”-t a meghajtóban. Használata egyszerű, a kiválasztott fájlt csak „mentés” vagy „drug-and-drop” a meghajtóba. Alfresco intelligens „ledobási helyeket” képzel el (Drop Zones), ahol a felhasználók egy mozdulattal „beledobhatják” a megosztott mappába a dokumentumaikat, ahol már automatikusan érvényes lesz rájuk a tartalomkezelő megkötései. Továbbá nagyon fontos megjegyezni, hogy a beadott fájl, (ha lehetséges) automatikusan végbemegy egy konverzió, ami egy semleges formátumra való alakítást jelent. Ilyen esetben vagy ODF vagy PDF lehetséges. A folyamat megengedi a Microsoft és/vagy OpenOffice használók számára, hogy megosszák az adataikat, ahol az egyedi formátumok, (mint például az Excel) át lesznek alakítva ODF-re vagy PDF-re. Ha már megtörtént a konverzió, akkor már automatikusan elküldhetők áttekintésre, ahol elbírálásra kerül.

Alfresco rekordkezelése

Sajnos a rekord kezelőt saját magam nem próbálhattam ki, ugyanis gondok vannak a kompatibilitással a 3.X verzióknál. A gond az, amit számos sikertelen próbálkozás után a fórum egyik témájában találtam meg, így szól: a rekordkezelő modul úgy néz ki, hogy nem települ fel rendesen a 3.X SP 1 Enterprise változatoknál. Ezidáig két hibát fedeztek fel. Így sajnos csakis a talált anyagokból tudok információkat szolgáltatni a rekordkezelésre. Közelebbi információkat a problémáról a <http://issues.alfresco.com/jira/browse/ETHREEOH-1422> oldalon találnak.

Egy állandó folyamat, amely a rekordokat egy semleges közeg alapján kezeli, megfelelően az eljárásokkal, ütemezésekkel és a megfelelő irányvonalakkal szemben. Egy rekord bármilyen dokumentum lehet, bármilyen tartalmi típusal, mint például Office dokumentum, weboldal, e-mail, videó, zene, együttműködés. A rekordkezelés egy tudományterület, ami hosszú időn keresztül megadja és definiálja a szabályokat rekord létrehozására, megtartására és törlésére. Visszatartási ütemterv az alapja egy befejezett rekordkezelési eljárásnak, ahol különböző rekord típusoknak különböző visszatartási irányelvei vannak, melyet a cég Szervezeti és Működési Szabályzatának megfelelően. A rekordkezelés megváltoztatta a dokumentumról al-

kotott nézeteket, abból a szempontból, hogy eddig egy egyedülálló dologként kezeltük, de most már egy átfogó életciklus fázisokkal rendelkeznek, mint:

- **Készítés: (creation)** Az Alfresco automatikusan kiemeli a meta adatokat a dokumentumokból, melyeket belehelyeztünk az intelligens virtuális fájl rendszerbe (IVFR), melynek segítségével kiterjesztett attribútumokkal is kibővíthetjük.
- **Osztályozás/ besorolás: (classification)** egy „tárgyat” és „címet” kell hozzárendelni egy rekordhoz, majd visszatartási ütemtervet adni. Dokumentumok az IVFR-ben automatikusan osztályozásra kerülnek és egy fájl tervet rendelnek hozzá.
- **Raktározás: (storage)** meg kellene, hogy őrizze a formátumot és egy semleges független tároló rendszerben legyen. Fájlrendszerben őrzi meg és raktározza a dokumentum formátumát
- **Hozzáférés irányítás: (access control)** A rekord láthatósági jogait felügyeli és azt, hogy ki mit eszközölhet rajta. A hozzáférés, irányítás a munkafolyamba van beleépítve, ami teljes hozzáférést biztosít és a rekord láthatóságát irányítja
- **Átnézés, megfeleltetés: (review, approval)** A megfeleltetés a hozzáférés és a rekord életciklusából kifolyólag is meghatározható. Példa: ha egy rekord létrehozási fázisban van, akkor csakis a létrehozó által látható
- **Naplózás:(auditting)** Egy naplózási fájlt kell vezetni minden eddigi eseményről ami a rekorddal kapcsolatos, ilyen lehet az, hogy ki és mikor frissítette a dokumentumot vagy az, hogy mikor lett megnézve. Átfogó naplózási lehetőségeket biztosít
- **Keresés és megtalálás: (search and discovery)** minden releváns rekordot meg kell, hogy kapjunk egy query-re, ezért kell, hogy kereshessünk:
 - teljes szövegre
 - kategóriára vagy tárgyra
 - attribútumra és

- o helyre

A rekordot a dokumentumtól az különbözteti meg, hogy közrejátzott-e üzleti döntésben. Ilyen eset lehet például: jelentés előkészítője, hivatalos esemény dokumentálása, jogi vagy gazdasági információ. A rekordokat céljuknak megfelelően szokták megnevezni, ilyen pl. adminisztratív, gazdasági, jogi, üzemeltetési, vagy történelmi. Minden rekord egy „tárgyat” egy „címet” és a saját megtartási irányvonalait tartalmazza. Nézzük meg, hogy az Alfresco rekord kezelője milyen lehetőségeket biztosít számunkra:

- Adminisztráció: adminisztrációs eszközök engedjék meg, hogy eseménynaplók, query-k és jelentések könnyen kidolgozhatók és futtathatók legyenek
- Automatikus életciklus váltás: (Automated Lifecycle Change) Annyit tesz, hogy dokumentum és metaadat változások aktiválását ütemezi be, egyszerű script-es szabályok alapján. Pl. ha a szerző befejezte a munkát metaadatokat ad meg, melyekre később keresni lehet, ezután a fájl átjut az átnézés fázisba, ahol automatikusan akár kiegészülhet egyéb, a szerző által nem feltüntetett meta adatokkal
- Automatikus ODF vagy PDF formátumba való konvertálás irodai formátumokból, ha bekerül a rendszerbe. Ezáltal szolgáltató semleges formátumot kap a dokumentum.
- DOD 5015.2 követelmények: USA Védelmi Minisztériumának (Department of Defense) 1997-ben kiadott szabványa. Azóta de facto szabványa lett sok kormányirodának. A szabvány egy átfogó rekord ütemezést ír elő, ami meghatározza az összes kormányirodának a rekorddal kapcsolatos politikáját, beleértve a dokumentum megtartást és eldobást.
- Elrendezés: (disposition) Irányított és előre meghatározott kezelést biztosít archívumokra, átcsoportosításokra, gyarapításra vagy törlésekre felhasználva a szabályokat és az automatikus feldolgozásokat. Meghatározza, hogy mi fog történni a rekorddal, megreked-e vagy megsemmisül vagy archiválásra kerül. Továbbá a munkafolya-

- Fájldervek: (fileplans) Strukturált tárgyra irányuló kitöltési sémát biztosít, amire szüksége van ahhoz, hogy egy megtartást ütemezhessen, így a tulajdon visszahívás és törlés kivitelezhető a megfelelő rekordokon. JavaScript segítségével a fájlderveket, dokumentumneveket és rekordtípusokat tudunk kezelni. Szabályok automatikusan életbe lépnek, ha egy ütemterv lépésének megfelel vagy akár metaadat változásnak. Ez a megrekedés és a dokumentum életciklusának az irányításánál fontos.
- Fogd és dobd belépés: (drag and drop entry) amely egy fájlrendszer nézetet nyújt, ahonnan egyszerűen csak kiválasztjuk a kívánt fájlt és beleillesztjük az Alfresco adattárterületére. Ezt egy szabványos Explorer segítségével megtehetjük. Evvel egyenes ágon kinyerjük a metaadatait és az új fájlok automatikus elrendeződnek. Ez mind a hozzájuk rendelt munkafolyam és fájldervek alapján történik.
- Megtartási és tárolási politikák: (retention and Archival Policies) Ez felel az elavult rekordok archiválásáért és a már archivált rekordok visszakereséséért, avagy végleges törléséért, amit a megtartási politikája határoz meg.
- Metaadat előre népszerűsítése: (pre-population of metadata) azaz előre definiált meghatározásokkal és sablonokkal automatikusan besorolhatunk és kezelhetünk rekord bejegyzéseket

- Metaadat érvényesítés és munkafolyam kezelés: (metadata validation and workflow handling) Egy modell irányított érvényesítéssel és kezeléssel tudjuk az új rekordok sértetlenségét biztosítani
- Rekordok kivetítése: (screening records) Előre meghatározott jelentések és metaadat típus definíciók arra, hogy, hogy keressünk és hogy vetítsük ki a rekordokat megfelelően a kezelési és kezelés kivételi esetekre
- Hozzáférés, megfeleltetés: A hozzáférés, irányítás a munkafolyamba van beleépítve, ami teljes hozzáférést biztosít és a rekord láthatóságát irányítja.

Összefoglalás:

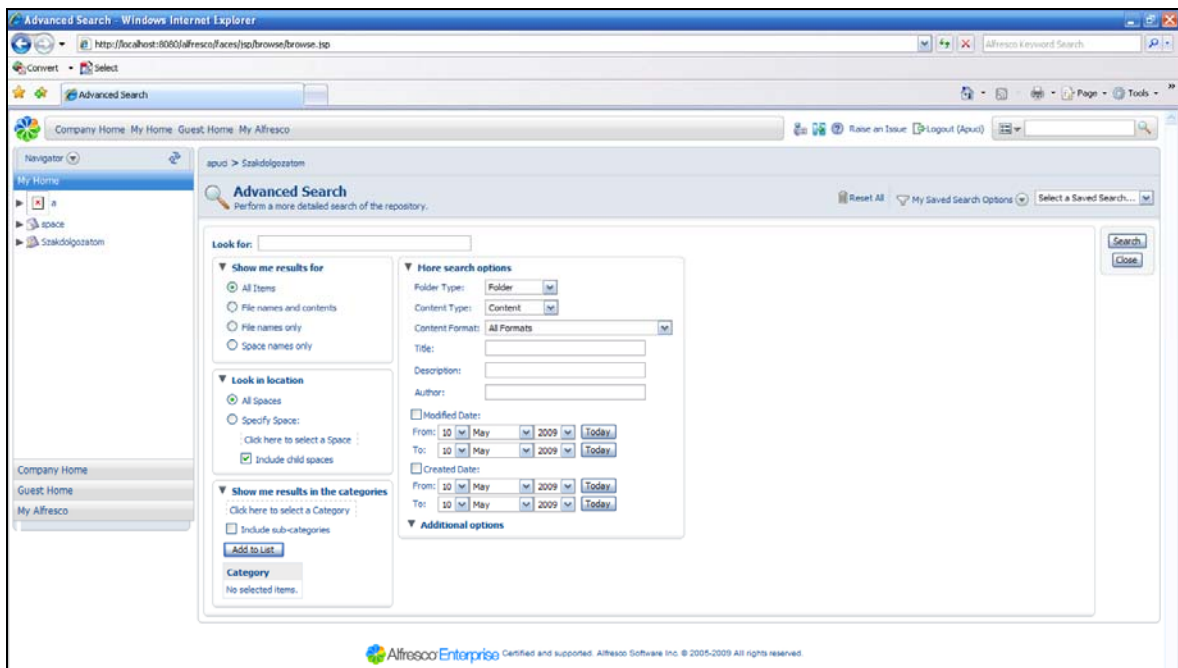
Az esemény vezérelt fájlrendszer könnyűvé teszi a kifinomult irányelvek beállítását. Sablonokkal oldják meg az egyes szabványos kitöltéseket, így magának a felhasználónak csak a szükséges dolgokat kell kitöltenie, a többit a rekordkezelő intézi. Ezzel leegyszerűsödik és minimalizálódik a hibalehetőségek száma, ugyanis minden el van rejtve a felhasználó elől. Neki már csak a sablonokat kell kitöltenie. Automatizált vízjegyek (watermark) hozzáadása a dokumentumokhoz, ODF és PDF konverzió, ami a mai csúcstechnológiát jelenti a tartalomkezelésben. Jó keresési mechanizmussal rendelkezik, egyesítve a meta adatokra, teljes tartalomra, a rekord helyére és kategóriájára való keresést (amit a keresés és megtalálás fázisban történik). Kereséseket menteni is lehet (erről részletesebben majd a keresésnél).

Alfresco keresési mechanizmusa

Hogy is keresnek napjainkban az emberek? Legtöbb ember semmilyen tudással nem rendelkezik keresés terén. Egyszerűen behívják a Google-t ahová beírják a keresett kifejezést, vagy részletét, majd eredményül kapnak egy találati halmazt. Ezt tovább böngészve remélhetőleg megtalálják a kívánt tartalmat. Olyan megoldásról, hogy a böngészők lehetőséget nyújtanak nyílt keresésekre (OpenSearch) már csak keveseknek van fogalma sajnos. Pedig a legnépszerűbb böngészők képesek az

OpenSearch-t támogató oldalakon való keresésre, egy egyszerű választólistából ki-keresve. Ilyen kereshető oldal lehet a blog, a wiki, az Amazon, mely mind támogatja a nyílt keresést. Ezt megcáfolandóan a statisztikák szerint pedig ez a legfelkapot-
tabb módja napjainkban a keresésnek. Miért is? Hisz a Google is a nyílt keresést támogatja. Ilyenkor feltehetnénk a kérdést. Hogy tehetné a tartalomkezelő szoftver a keresést oly egyszerűvé, mint a Google? Erre a választ a nyílt keresésben találhatjuk meg.

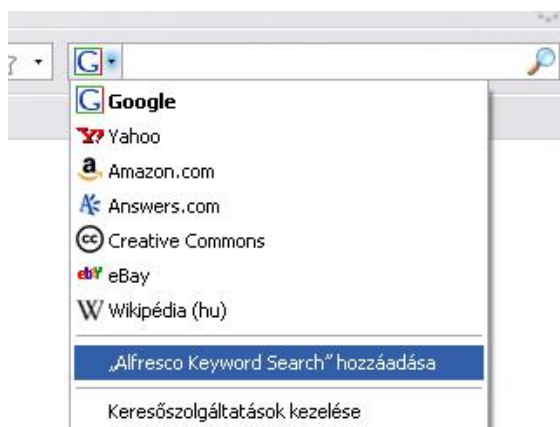
De először is vizsgáljuk meg, hogy egy átlagos munkás mennyi időt tölt ke-
reséssel, és ennek a cégre való hatásait. Azok az emberek, akik sokat keresnek körülbelül 150 órát töltenek el kereséssel évente. A „keveset keresők” már csak az 1/8-át, azaz úgy 30 órát évente. Ennek két oka lehet: 1. vagy nehéz a keresés 2. vagy a munkatárs nem ért hozzá. Azért feltételezzük, hogy jó képzést kapott, és így csak a keresést lehet hibáztatni. Ezen felül, ha jobb és könnyebb keresést tudnánk biztosítani, akkor akár az idejük 50%-át is megtakaríthatnánk, amit haszno-
sabb munkára is fordíthatnának. Emlékezzünk vissza, hogy mennyi pénztől esik el a vállalat ilyen „kis” hibákból adódóan, mint amit a dokumentumkezelésnél említett két kutatás is kimutat. Ezért meg kell vizsgálnunk, hogy milyen követelményeik vannak a felhasználóknak a kereséssel szemben: Első nagy gondot az lehet, hogy belső keresés nehézkes. Nehezen találják meg a helyes információt, akár a saját cégükön belül is, sőt némelyek az is mondják, hogy könnyebb egy idegen vállalatról információkhoz jutni, mint a sajátjához.



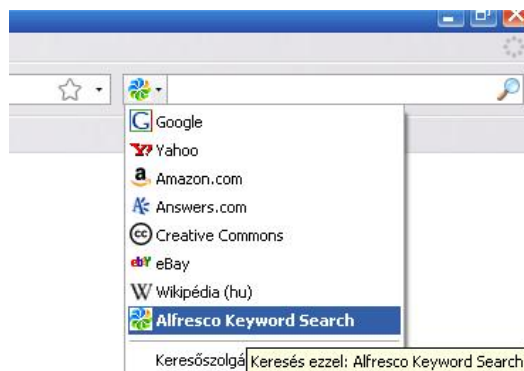
13. kép: összetett és egyszerű keresés

Második az, hogy a felhasználók több mint a fele több forrást is meg kell hogy vizsgáljon, hogy egy átfogó képet kapjon az információról. Tovább nehezíti a keresést az, ha megtalálták a helyes dokumentumot, akkor annak melyek verzióját kell használni. Melyik az éppen érvényes, a használatban lévő, ugyanis nem előírás, hogy a legújabb dokumentum legyen az. Ennek eredménye, hogy a felhasználók 42%-a véletlenül rossz adatot használ fel egyszer/ hét, ami rávilágít arra a tényre, hogy a keresés eredményessége szempontjából a helyes dokumentum megtalálása mellett a jó verzió kiválasztása is ugyanolyan fontos.

Most nézzük meg azt, hogy egy keresés hogyan működik egy dokumentum kezelő rendszerben. Igazság szerint kereséskor egy dokumentumra keresünk, de valójában a dokumentum tartalma az ami érdekel minket, függetlenül a fizikai vagy éppen formai megjelenésétől. Egy dokumentum tartalmára kereshetünk, pl. kulcsszavas keresés vagy teljes szövegre, de ezen belül egy dokumentumnak van még egyéb tulajdonsága is. Egy Word dokumentum gyakran egy sablonnal együtt érkezik, ami a szerzőt, közreműködőket, kulcsszavakat és a címet tartalmazza, ami a munkások számára szintén kereshetővé kell tenni. Keresni a már említett nyílt keresési lehetőséggel tudunk.



14. kép: Alfresco adattárhely hozzáadás



15. kép: adattárhely hozzáadva

Nyílt keresés

Egy de facto szabvány, amit az Amazon és az A9³⁰ vezetett be 2005 őszén. A nyílt keresés³¹ egyszerű formátumok gyűjteménye, keresési eredmények megosztására. Bonyolultabban: technológiák gyűjteménye, melyek képesek a keresési eredmények egyesíthető és automatikusan frissülő formátumra való konvertálásra és egyidejű publikálásra. Ami arra hivatott, hogy az Atom és RSS³² kombinációját helyettesítse. A nyílt keresést leíró dokumentum (OpenSearch description dokument³³) formátum használható arra, hogy egy kereső motort leírjon, és így használhatóvá váljon a keresést kezdeményező kliensalkalmazások számára. Ilyen alkalmazások a böngészők, például IE és Firefox ahol az Explorer támogatja a nyílt keresésre utaló kivételt (OpenSearch Referrer Extension³⁴) és a Firefox pedig a nyílt keresést javasoló kivételt (OpenSearch Suggestions Extension³⁵). Ahol az első a kereső motoroknak megengedi, hogy a böngésző keresési adatait lekérjék, ahhoz hogy jelezzék a keresés forrását. Az adatokat arra használhatnánk, hogy egy bizonyos kereső klienst beazonosítsunk, annak verzióját és egyéb információkat kapjunk a

³⁰ <http://en.wikipedia.org/wiki/A9.com> 2009.05.06.

³¹ <http://www.opensearch.org/Specifications/OpenSearch/1.1> 2009.05.06.

³² <http://www.rssboard.org/rss-specification> 2009.05.06.

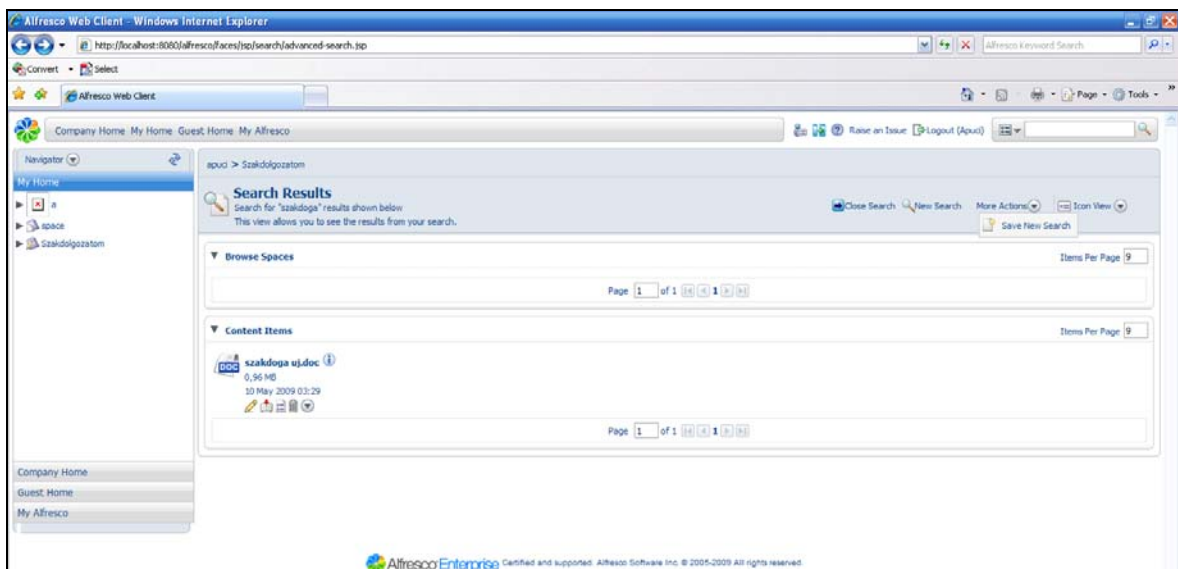
³³ http://www.opensearch.org/Specifications/OpenSearch/1.1#OpenSearch_description_document

2009.05.06..

³⁴ <http://www.opensearch.org/Specifications/OpenSearch/Extensions/Referrer/1.0> 2009.05.06.

³⁵ <http://www.opensearch.org/Specifications/OpenSearch/Extensions/Suggestions/1.0> 2009.05.13.

keresés forrásáról. Ilyen adat lehet pl. a verzió (version), amit az IE7 így jelezhetne például `<source>IE7</source>` vagy esetleg `<source>IE v.7</source>`. A nyílt keresést javasoló kivétel pedig egy szabályt / előírást nyújt, amelyben a kereső motorok egy keresési fogalmak halmazát adnák vissza, amelyben egy adott, keresendő prefixum további kiegészítő fogalmait kapnánk vissza. Ezután a befejezett keresések használhatóvá válnának egy keresési kliens számára, ahhoz hogy dinamikusan szolgáltatassák a végső felhasználóhoz a keresési szavak javaslatát. Erre a legjobb példa a Google suggest³⁶, ahol a kereső felület a felhasználóval való interakciója közben a keresőnek folyamatosan szolgáltatja a kereshető további lehetőségeket. Az igazi újítása ott van, hogy az Internet mára már úgy kinőtte magát, hogy egyetlen kereső robottal vagy keresési algoritmussal nem lehet mindent kikeresni, sőt, a kereső robotok a tartalom csak kis hányadát észlelik. A legjobb keresési motorok azok, melyek olyan emberek számára íródtak, akik a legjobban ismerik a tartalmukat.



16. kép: összetett keresés eredménye és keresés mentése

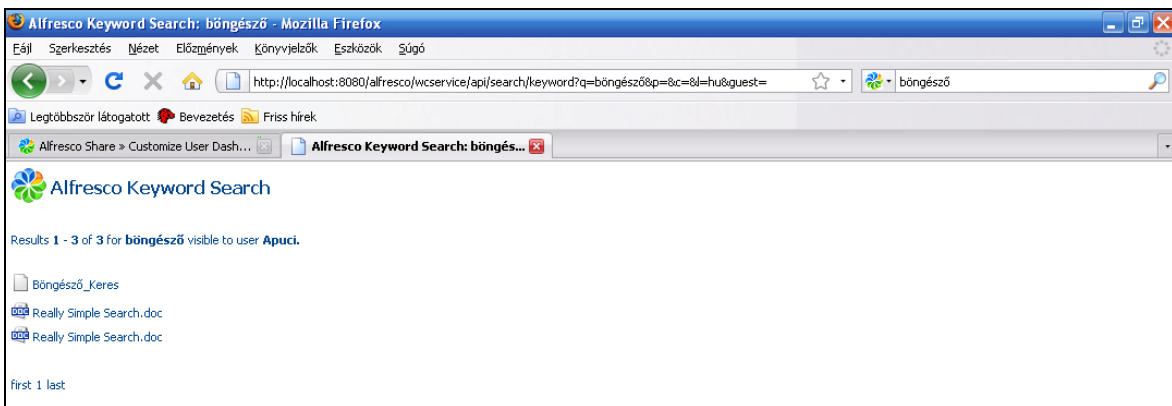
Erre most szkeptikusan feltehetnénk a kérdést, hogy a régi jól bevált adaperet (perifériaillesztő) miért cseréljük le nyílt forrású keresésre. Jogos lenne a kérdés, de ha belegondolunk, akkor egy adattárhelyen való keresés azt jelentette

³⁶ <http://labs.google.com/suggestfaq.html>

régen, hogy több belső tárhelyen való keresést végzünk, mint például FileNet vagy Documentum rendszerekkel. Ebben az időben a megoldást perifériaillesztők jelentették, napjainkban ezt is jelenti, de kiegészül még külső adattárhelyek elérésével, keresésével is. Mivel a perifériák erre nem képesek, ennek megfelelően a perifériák használata tarthatatlan, megengedhetetlen lett. Ugyanis a perifériaillesztőket állandóan naprakész állapotban kell tartani, ahhoz hogy használni is lehessen őket. Az pedig nagy feladat abból kiindulva, hogy mekkorák, és milyen gyakoriak a keresések. Ha a szabványt követjük, a nyílt keresést, akkor a jövő felé tekintünk, ahol egyszerűbben, gyorsabban és precízebben kaphatunk eredményt. Alfresco természetesen támogatja, sőt erre építi a kereséseit. Ő az automatikus felfedezéssel (Auto Discovery) oldotta meg azt, hogy a böngészők Alfresco kompatibilisek legyenek, vagyis kereshessünk az adattárterületében. Használatához mindössze annyi a dolgunk, hogy elindítjuk az Alfresco Explorert, és hozzáadjuk a már létező keresőkhöz az Alfresco keresőjét. Ezután már a jól megszokott módon lehet keresni. Ezen belül automatikusan nyelvfelismeréssel rendelkezik, ha például angol böngészőt használunk, akkor angol dokumentumokat kapunk első megközelítésben. A visszakeresés megkönnyítéséhez teljes szöveges indexelési rendszert (Full Text Indexing System) használ. Ennek az előnye, hogy fájl tulajdonságokat is tartalmaz, pl. Word fájl szerzője, keletkezésének dátuma stb. Alfresco ezt relációs adatbázis kezelőben tárolja. A felhasználónak lehetősége van indexelni a dokumentum tartalmát, ezáltal nem csak a tulajdonságokra, hanem egyéni címkékre is kereshet majd. Tartalom visszakeresésére a Lucene³⁷-t használják és relációs adatbázis kezelő rendszereket attribútumok visszakeresésére használnak. Lucene egy keresőmotor, ami alapvetően dokumentumokat kezel, melyek részhalmazát ki kell tudni választani egy keresési feltétel alapján. A dokumentumokat mezőkre bontja és a keresés eredményét rangsorolja, majd megadja, hogy a keresésre mennyire releváns a dokumentum.

Ahhoz hogy átfogó képet kapjunk egy témáról több forrást is figyelembe kell vennünk. Ez azt jelenti, hogy külső információra is szükségem lehet akár, amit már nem biztos hogy egy Afresco adattárhelyéről tudni beszerezni, így a különböző adattárhelyeket egyeztetni kell. Erre is ad lehetőséget a nyílt keresés szabványa.

³⁷ <http://en.wikipedia.org/wiki/Lucene> 2009.05.18.



17. kép: böngésző keresés eredménye

Keresés és biztonság

Az egész arról szól, hogy a keresést és a biztonságot összeolvasszák. Sok esetben megtörtént, hogy egy belső keresési rendszert telepítettek, de olyan jól sikerült, hogy senki sem tudott keresni, ugyanis nagyon pontatlan találatok születtek. Ezért a legmegfelelőbb eset az, ha elégséges biztonsági szintek vannak életben. Kereséskor minden releváns dokumentumot megkapják, vagy ha arra nincs jogosultságom, akkor ne is jelenjen meg a képernyőn, még akkor sem, ha keresésemre megfelelne. Ellenben a biztonság nem állandó, változik az idő folyamán. Ezt nevezük életciklusnak (rekordkezelésnél leírtak érvényesek).

Ha böngészek az Interneten, és nyílt keresést alkalmazok az Alfresco keresővel, akkor is biztonságos keresést végzek. Vagyis ennek feltétele az, hogy amint kiadtuk a keresésre a parancsot, az Alfresco megkér minket, hogy jelentkezünk be. Szerencsére csak az első alkalommal. Ekkor egy biztonsági profilt kapunk, amivel már bizonyos dokumentumokhoz hozzáférhetünk, a dokumentumok életciklusából adódóan. Ebből következik, hogy nyílt keresés vagy Alfresco web kliens keresés, mindegy, mert mindkettővel ugyanazokat a biztonsági szintek érhetők el.

Enterprise Social Software

Itt a Social Software trendjeiről lesz szó az intézményekben, hogy mit is jelent valójában a Web 2.0. és hogyan lehet elérni a legjobb támogatást a tartalom kezelési stratégiákban.

Web 2.0 és social szoftvernek nagy befolyással bír a tartalomra, a dolgozók termelékenységére, belső architektúrára és a szabványokra egyaránt. Ez például a keresésnél jelenik meg, abban hogy hol keresem a dokumentumokat, vagy a saját belső dokumentumok között vagy Google. Nincs többé éles határ a külső és belső dokumentumok között, hisz hányszor kell egy külső forrásból információhoz jutni. Tudásmenedzsereknek mindkettőt kell használniuk, hogy hatásosan dolgozzanak.

A vezető vállalatok nem csak egy statikus web rendszert szeretnének maguk köré fölépíteni, hanem közösségi oldalakat a termékeik dinamikus bemutatása köré is. Ilyen Web 2.0 lehetőség a blog, a wiki és a fórum, ahonnan visszajelzéseket kapunk vélemény nyilvánításokon keresztül. Itt szavazhatnak, rangsorolhatnak, jelzőkkel láthatják el a tartalmat. Ezáltal a közösség „véleménye” oly fontossá válik, mint a cég testületi érdekei. A felhasználók új módon szeretnék megtalálni az általuk keresett tartalmat, mint például jelzeteken keresztül, amelyek megmutatják, hogy mely hozzászólások a leghasznosabbak, melyek voltak a barátaik hozzászólásai és hasonlók. Ilyen rendszerek beiktatása állományunkba igazi piaci helyzetelőnyhöz vezethet.

A social software egy REST³⁸ (Representational state transfer) nevezetű megjelenítésre építették. REST lényege az, hogy minden megcímezhető URL segítségével, mint a tartalom, mint az alkalmazások is. Social software még a további új szabványokat is felhasznál: RSS, Atom publishing³⁹, OpenSearch, OpenSocial⁴⁰.. Továbbá a social software gazdag internet alkalmazásokkal van felszerelve, ilyen pl. az Adobe⁴¹ termékek vagy AJAX..

A Social Software legfontosabb hatásai

³⁸ http://en.wikipedia.org/wiki/REST	2009.05.02.
³⁹ http://tools.ietf.org/html/rfc5023	2009. 05. 13.
⁴⁰ http://en.wikipedai.org/wiki/OpenSocial	2009. 05. 13.
⁴¹ http://www.adobe.com/products/	2009.05.13.

Az a tény, hogy a tudásmenedzsereknek manapság már nem csak belső hanem külső forrásokból is kell, hogy anyaghoz jussanak, arra készíti a cégeket, hogy újra értékeljék a tartalomkezelést, és szolgáltatására való stratégiájukat. A Social software segítségével, sokkal egyszerűbben, kényelmesebben és termelékenyebben lehet tartani a kapcsolatot a munkatársainkkal és/vagy a felhasználókkal. Ami régebben úgy tűnt, hogy túl összetett, az most már felhasználóbarátabb lett. Kiegészült nagyon sok kellemes és hasznos lehetőséggel, ami pl.:

- Social Profiling (Társadalmi csoportosítás): a felhasználók elvárják azt hogy képesek legyenek arra, hogy saját profilt készíthessenek az ő érdeklődésükről, tapasztalatukról és háttérükről. Jó példa: Facebook⁴²
- Social Graph (társadalmi gráf): elvárják, hogy könnyedén kapcsolatba lépjenek barátaikkal, munkatársaikkal, és hasonló érdeklődésű csoportokkal. Jó példa a LinkedIn⁴³
- Social Content: A felhasználók elvárják azt hogy egyszerűen készíthessenek tartalmat egy weboldalon és hogy olvashassák. Más felhasználó véleménye olyan fontos lesz mint a közösségi tartalom. Erre jó példa a Wikipédia és a Youtube
- Social Feeds and Personalization: A felhasználók elvárják azt, hogy könnyen beállíthassák a saját érdeklődésüket az RSS adásokon keresztül és automatikusan jelezzék számára, ha az az információ megváltozott.
- Social Tagging (társadalmi címkézés/jelzeteles): a tartalom egyszerű címkézését enged meg
- Social Commenting (társadalmi hozzászólások): Manapság a hozzászólások már oly fontosak mint a tartalom maga, vagy már fontosabbak is. A felhasználók elvárják, hogy könnyen tudjanak hozzászólni bármihez az Interneten. Digg⁴⁴ a jó példa erre
- Social Rating (társadalmi rangsorolás): a felhasználók elvárják, hogy egyszerűen rangsorolhassanak tartalmat, és ez alapján kereshessen más is.

⁴² <http://www.facebook.com/>

2009.05.13.

⁴³ <http://www.linkedin.com/>

2009.05.03

⁴⁴ <http://www.digg.com/>

2009.05.13.

- Social Search (társadalmi keresés): ne csak kulcsszavakra lehessen keresni, hanem társadalmi információkra, mint például címkékre, legtöbbet szavazott, legtöbbet letöltött. Ezt még fejleszteni lehet azzal, hogy hozzátesszük azt, hogy a barátaink mit néztek, mire szavaztak és mit töltöttek le. „A tömeg tudása” alapú keresés
- Social Mobility (társadalmi mobilitás): Internetes oldalak egyszerű elérése a saját PC-jükről vagy mobil berendezésükről

Felsoroltak közül a cégek működéséhez nem kellenének blogok, wikik se fórumok, csakis a vásárlók és a partnerek miatt szolgáltatják. A WCM megoldásoknál melyek a Web 2.0 előnyeiket kihasználják nem a biztonság és a társadalmi irányítás a nehéz, hanem az életciklus kezel, vagyis a dokumentum életútja. Azaz, ha egy dokumentum életciklusa elérte a publikus szintet, akkor az már maradjon ebben az állapotban, és a rá érvényes jogok is maradjanak állandók, amíg a tulajdonosa meg nem változtatja azt. Social Software felfogásban a közösség irányítja a tartalmat, ő dönti el, hogy a jó tartalom kedvezőbb, míg a rossz hátrányosabb helyre sorolódjon, vagy akár teljesen törlődjön. Az előbbi tevékenységet megnehezítik a vállalatnak az ellenőrzési és felügyeleti elvárásai. Egy dokumentum kezelőben a dokumentumokra életciklusokat állíthatunk be, ahol egy ciklus a következőkből áll.

Dokumentum életútja: privát-> jóváhagyott-> belsőleg elérhető-> publikus. Ezután már lehet rangsorolni, címkézni, és véleményt nyilvánítani róla. Vállalatoknál ez annyiban másabb, hogy ők a projektekhez keresnek új munkásokat, vagy már meglévő projekteket keresnek ki, és azok tervezőit keresik fel, hogy csatlakozzon hozzájuk. Ez egy olyan modellt követel, ami nem csak emberekre, hanem tartalomra és projektekre is kiterjeszhető. Cégek profitjának növelésének a legjobb lehetősége napjainkban a termék beállítottságú közösségi oldalakban és a belső együttműködésben keresendő, melyek a terméket, nem a vállalatot propagálják. A Social Software egyedre és azok barátaira összpontosít. Ahhoz, hogy egy vállalat kijelenthesse, hogy ő igenis, Social Software használó, tennie kell. Egy wiki használata az oldalon nem teszi rögtön a cégünket vezető Social Software vállalattá. Ehhez módosításokra van szükség:

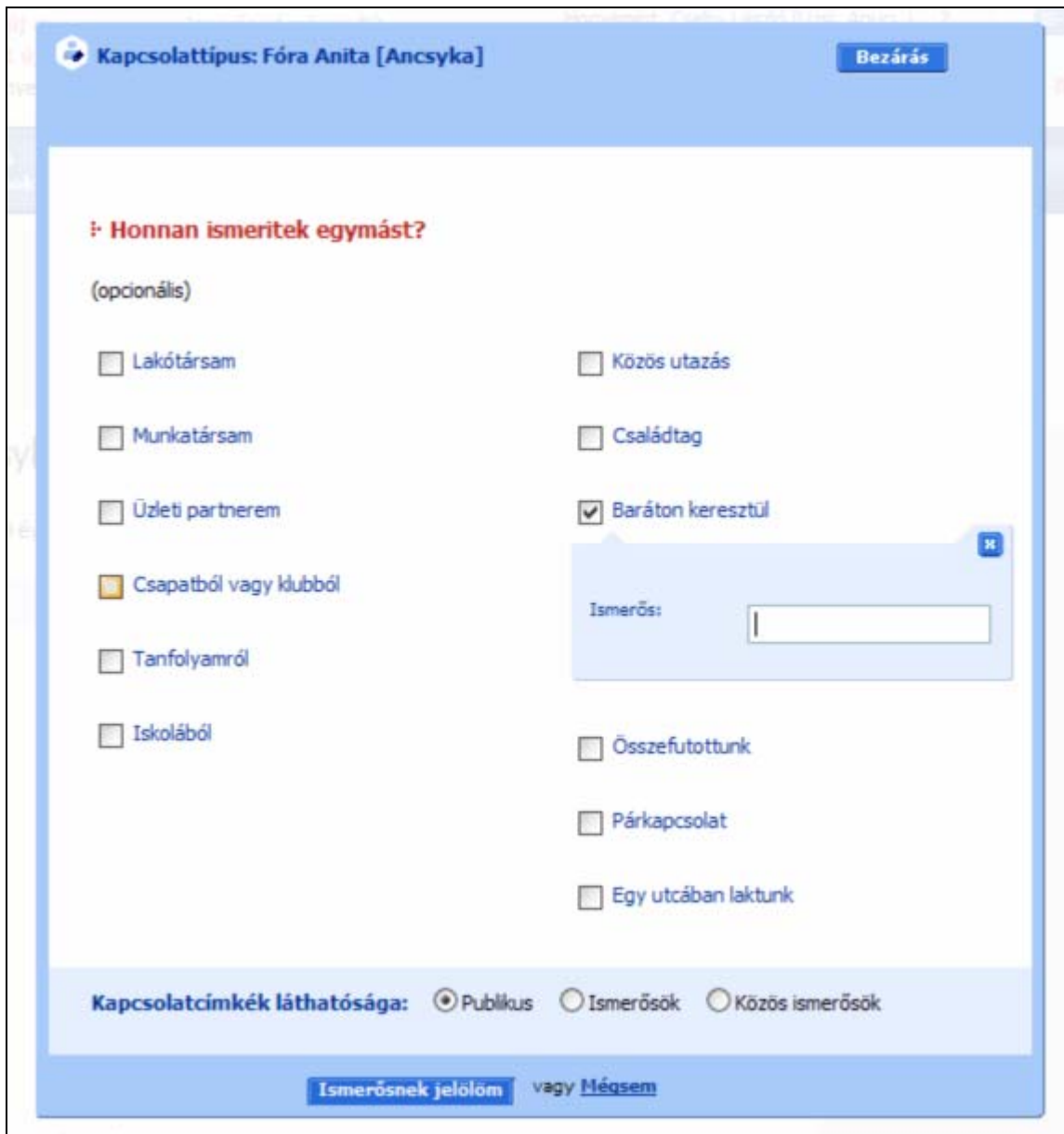
- helyettesítsük a megosztott meghajtót egy portálra, mivel a legtöbb tudás úgyis Word, Excel és PowerPoint dokumentumokból származik
- kell egy olyan program ami embereket, projekteket és tartalmat kezel, és ezt RSS és Atom segítségével szolgáltatja
- közösségi szoftvere: termék orientált közösségi oldalakat kell megtervezni

Ha ez mind sikeres volt és alkalmazzuk akkor még azt kell figyelembe vennünk, hogy az Enterprise Social Software számára kulcsfontosságú a REST architektúra. Ez engedélyezi, hogy az összes query, tartalom, meta adat és felhasználói felület összetevő egy URL megadásával elérhetővé váljon.

Nyílt forrású szoftver és hasonló innovatív architektúra az, amivel a vállalat megtakarítása, használhatósága és hatékonysága növelhető. Ilyen jól bevált nyílt forrású program például a Linux, MySQL, Tomcat és PHP. Ezek a technológiák használhatók a vállalati programok mellett, melynek eredményeként, több felhasználó fogja használni a mi oldalainkat, ami viszont több potenciális vásárlót jelent. Egy jó social software-nél mindig minden oldal egyénre szabott. Keresésnél fontos tényező például az, hogy a kollégánk mit is sorolt be magasan, és így csak a fontos részekre kell odafigyelni és nem az egész dokumentumra. Ha a belső és/vagy külső dokumentumra keresünk, akkor teljes mértékben kihasználhatjuk az OpenSearch keresési lehetőségeit.

A Social Software a népszerűsége, a címkéken és azon a tényezőn alapszik, hogy ki olvasta és értékelte. Ezeknek az információknak ugyanolyan súlya van, mint magának a tartalomnak. A népszerűséget az oldal használatából, értékelésekből, rangsorolásokból állapíthatjuk meg. A címkéket a végső felhasználók adják meg, és a rangsort „felhők” által lehet megadni. A kedvelt címkék nagyobb felhők lesznek, amíg a kevésbé kedveltek kisebbek. A társadalmi keresések nem csak azon alapulnak, hogy ki mit olvasott, értékelt, hanem azon is, hogy esetleg az a barátunk vagy egy hasonló ízlésű ember volt. Ez egy olyan tartalom modellt kíván, mely a tartalmat figyeli, attól függően, hogy ki olvasta, kapcsolatban vannak e velünk, milyen típusú ez a kapcsolat, hányszor lett elolvasva, és ezen felül egy hatásos eljárást biztosít, hogy ezeket az adatokat szemléltessük és elérhessük. Egy magyar példa

erre az új IWIW⁴⁵ lehet, ahol pl. új kapcsolat létrehozásánál, felkínál lehetőségeket egy barát tipizálására mint: munkatárs, családtag, barátan keresztül ismerem stb. ahogy azt a 18. kép szemlélteti. Jó esetben ezt RSS segítségével is elérhetővé teszik. Az RSS arról szól, hogy könnyen megtalálhassuk a tartalmat abban a formában amiben szeretnénk, összegezhessük azt ahogy szeretnénk, és értesítve legyünk a tartalomról, ahogy azt elvárnánk. Egy alapvető feltétel, hogy a rendszer különálló összetevőkből álljon, hogy a program minden pontja figyelhető legyen.



The screenshot shows a web form titled "Kapcsolattípus: Fóra Anita [Ancsyka]" with a "Bezárás" button in the top right. The main heading is "Honnan ismeritek egymást?" (Where do you know each other?). Below this, it says "(opcionális)" (optional). The form contains two columns of radio button options:

- Lakótársam
- Munkatársam
- Üzleti partnerem
- Csapatból vagy klubból
- Tanfolyamról
- Iskolából
- Közös utazás
- Családtag
- Baráton keresztül (checked)
- Összefutottunk
- Párkapcsolat
- Egy utcában laktunk

Below the options, there is a section for "Ismerős:" with an input field. At the bottom, there is a section for "Kapcsolatcímkék láthatósága:" (Relationship tag visibility) with three radio buttons: "Publikus" (selected), "Ismerősök", and "Közös ismerősök". At the very bottom, there are two buttons: "Ismerősnek jelölöm" and "vagy Mégsem".

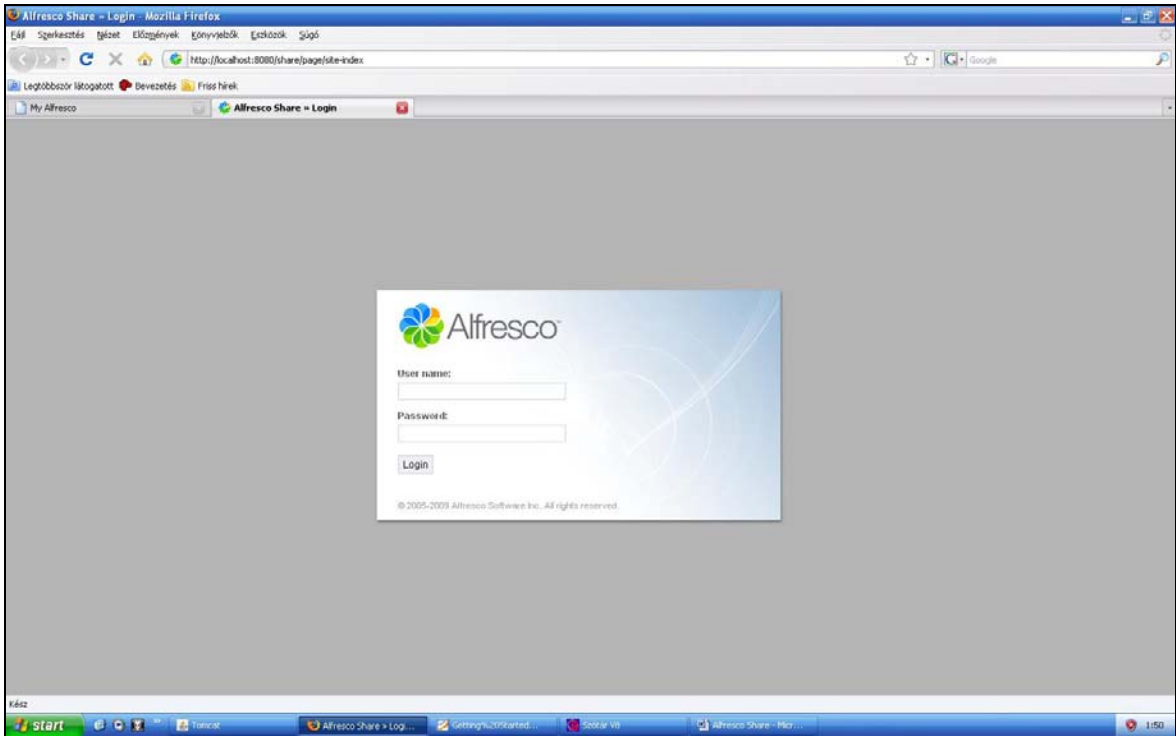
18. kép: iwiw felhasználói kapcsolat meghatározása

⁴⁵ <http://www.iwiw.hu> 2009.05.03.

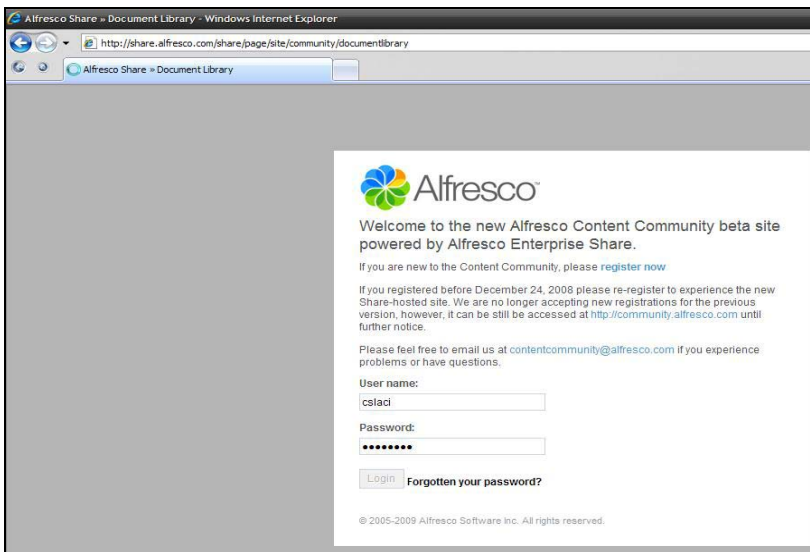
Ahhoz, hogy külső és belső erőforrásokból összesíthessük a tartalmat, URL alapú hozzáférés szükséges. Az összefűzés nem csak két tartalom bemutatása egy böngésző két ablakában, hanem a két tartalom egyesítése és ebből egy harmadik létrehozása. Erre egy jó példa: ha az ember korát és keresetét vesszük, akkor megkapjuk a várható nyugdíját. URL-ek használatával tudjuk egyértelműen beazonosítani és elérni a webes tartalmakat. Ha külső és belső tartalmat szeretnénk egyszerűen összefűzni, akkor az adattárhelyünkre URL alapú hozzáférés szüksége. Egyszerűvé kell tenni a felhasználóknak, hogy visszajelzéseket tudjanak adni a szolgáltatott tartalmunkra. Ezért a megjegyzésekre, rangsorolásokra, szavazásokra, az egyértelműsége (ki írta) kell hangsúlyt fektetni, de éppoly fontos az is, hogy ki olvasta és mit gondol róla. Ez a Web 2.0 stílusú megközelítés. Alfresco ezt a Share nevezetű portál segítségével valósítja meg.

Alfresco Share

Mi is a Share? Egy olyan közeget biztosító programok egységes neve, mely lehetővé teszi különböző dokumentumok kezelését, rendezését és szolgáltatását, felhasználóbarát módon, egyszerűen és a már megszokott módszerekkel. Share leegyszerűsíti az információ befogadását, megosztását és visszakeresését virtuális csapatokon keresztül. Továbbá növeli a teljesítményt és csökkenti a hálózat sávszélességének az igényét. Működésének feltétele, hogy a szerver fusson, majd a programot elindítva, böngészőnkben megjelenik az Alfresco Share Login.



19. kép: belépés saját gépem az Alfresco közösségi oldalra

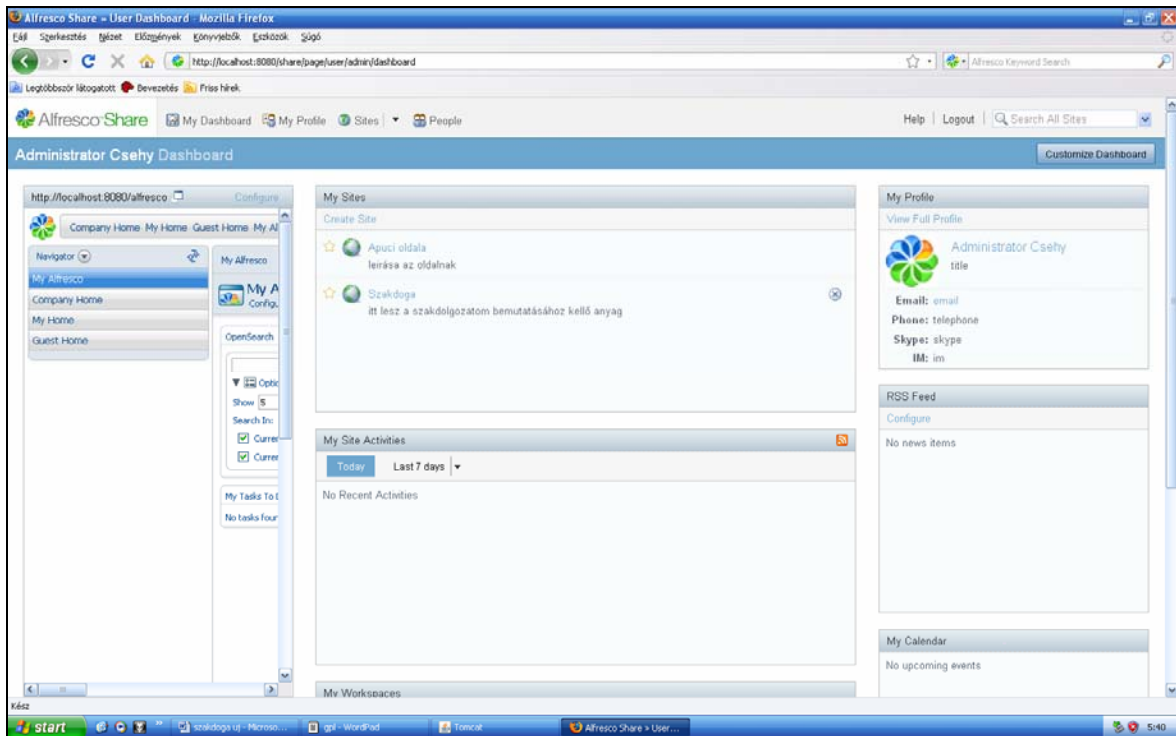


20. kép: internetes belépés az Alfresco Share dokumentum könyvtárába

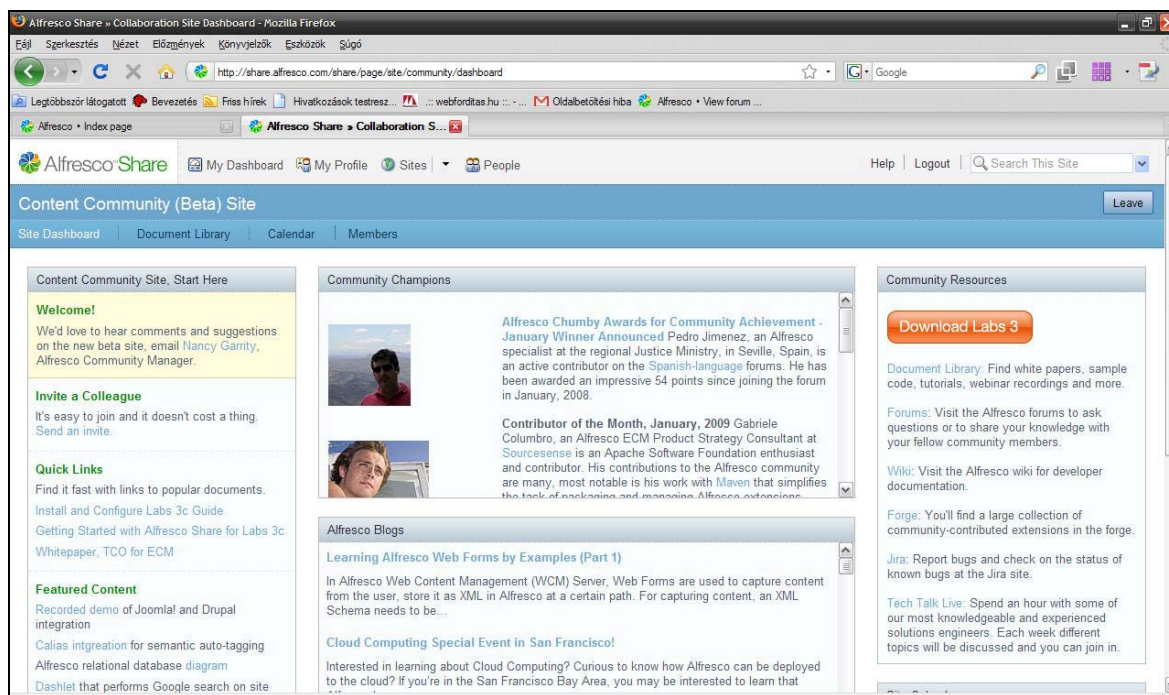
Első indításkor, az admin felhasználói névvel és jelszóval tudjuk azonosítani magunkat. Sikeres belépés után megjelenik a műszerfalunk (*munkafelület*) / dashboard, ahol az összes szükséges információt egyszerre megjeleníthetjük. **Dashbord** célja az, hogy automatikusan megjelenítse a felhasználónak a számára

⁴⁶ <http://share.alfresco.com/share/page/site/community/dashboard>

hasznos fájlokat, továbbá fontos elemeket, amikkel a napi munkája során gyakran találkozunk és használjuk, és mindezt felhasználó barát módon tegye. Evvel kimentenek minket a felesleges keresések alól, és időt spórolnak meg. Alfresco-nál a dashboard további kisebb úgynevezett dashletekből áll.



21. kép: Alfresco Share műszerfala, jól kivethetők a Dashlet-ek



47

22. kép: Alfresco Share internetes műszerfala

A dashlet egy összegző komponens, amely rendelkezik vagy nem egy meg-egyező oldal elemmel. Ha rendelkezik az oldal ilyen kiegészítő elemmel, akkor a kapcsolódó dashlet az innen kapott információkat jeleníti meg. Dashletek csakis a *személyi* avagy a készített oldalak dashbordján (Site Dashboard) jelennek meg. Készített oldal alatt azt értem, hogy bárki aki jogosult rá, a saját oldalán kívül, ami a személyi oldala, azon kívül készíthet más oldalt együttműködés, információcsere, megvitatás, wiki, blog vagy akár naptár céljából is. Alfresco számos ilyen dashlet-et nyújt, ilyen a:

- Alfresco Network: Alfresco felhasználóinak a portálja, arra, hogy mindig naprakész, a legújabb híreket, információkat közvetítsék felénk. Továbbá a hálózaton történő legújabb eseményekről is összegyűjt a híreket. Alapból látható.
- CMIS48 Feed: olyan linkek gyűjt össze, melyek a CMIS előírásaival, és használatával kapcsolatosak. Alapból nem látható.

⁴⁷ <http://share.alfresco.com/share/page/site/community/dashboard>

2009.05.10.

⁴⁸ Content Management Interoperability Services

- Documents I'm Editing: az összes dokumentumot kilistázza, melyeken változtatást viszek végbe. Vagyis check out az állapota. Alapból nem látható.
- Getting Started: arra nyújt lehetőséget, hogy jobban megismerjük a Share lehetőségeit, de amint megismerkedtünk a környezettel, akár törölhetjük is. Igazából első lépésekben segít. Alapból látható.
- My Calendar: összefoglalást nyújt az általunk készített eseményekről, és azokról melyek azokon az oldalakon történnek, melynek mi magunk is tagjai vagyunk. Alapból látható.
- My Profile: ez egy kivonata a profilunknak, alapból a nevünket, email címünket, telefonszámunkat, Skype és IM azonosítónkat emeli ki a teljes profilunkból. Alapból látható.
- My Sites` Activities: azokat a legújabb eseményeket figyeli és majd közli, melyek azokon az oldalakon történtek, aminek tagjai vagyunk. Alapból látható.
- My Tasks: minden feladatunkat az összes oldalról kimutatja. Alapból látható.
- RSS feed: alap beállításként a Global Alfresco RSS hírfolyam egészét szolgáltatja, ez a <http://www.alfresco.com/feeds/all> URL címen érhető el. Alapból tíz hírt jelenít meg, de természetesen mint a többi dashboard-et is, ezt is kedvünkre szabhatjuk, hogy a saját híreinket mutassa akár. Alapból látható.
- Web View: számunkra fontos weboldalt tudunk ide beállítani, ami mintha egy frame-ként jelenne meg az oldalon. Itt csakis www oldalakat tehetünk meg, sajnos mivel alapból úgy van beállítva, hogy mi csak a http:// protokoll után következőket írhatjuk be.

Egyéni műszerfal létrehozása

Alfrescon belül szinte mindent a saját elképzelésünk alapján tudjuk beállítani. Ez a műszerfal beállításánál is érvényes, ahol a megjelenítését (egy, két, három vagy akár négy oszlop áll rendelkezésre) és a dashboard-eket, hogy melyik és hol jelenjen meg. Ehhez a Customize Dashboard gombra kell kattintani, majd kiválaszthatjuk, hogy hány oszlopból álljon, és az egyes oszlopok mit, milyen sorrendben

tartalmazzanak. Akár mindent megjeleníthetünk, vagy esetleg semmit sem, csak az a kikötés, hogy egy oszlopban maximum öt elem lehetséges, annyit enged alapból.

Együttműködési oldal készítése

Ahhoz hogy a csapatunk a lehető leghatásosabban tudjon együttműködni, bárholnan a világon, szükség van egy közös pontra, ahol mindenki megoszthatja a munkáját, véleményét, előre tudnak egyeztetni időpontokat, találkozókát, írhatnak blogot, online készíthetnek új dokumentumokat, és ezt megoszthatják akár wiki segítségével. Erre is nyújt lehetőséget a Share. Ez az úgynevezett együttműködési oldal, ahol a csapatunk képes mindarra, amit előbb felsoroltam.

Létrehozásához a *saját* (személyi) műszerfalra kell mennünk, és ott a Sites gomb mellett van egy lefele mutató nyíl, és ott a Create Site pontra kattintva létrehozhatunk egy oldalt. Itt adatként bekéri az oldal nevét mintámban Szakdoga, majd az URL rövid kivonata (ami nem tartalmazhat szóközt vagy speciális jeleket), majd egy rövid leírás, majd az oldal típusa Collaboration Site (ahogy olvastam, a 3.1 verziónál csak ez van) és a végen a jelölődoboz, ami ha bepipált, akkor publikus oldal lesz, ha nem akkor marad privát.

Együttműködési oldal beállítása

Az itteni műszerfal minden információt és eseményt kimutat, ami az oldalhoz van kapcsolva, és az oldal gazdája tud beállításokat eszközölni rajta. Ilyen lehetőség:

- Az oldal műszerfalának beállítása dashletek segítségével
- Az oldal egyedivé tétele avval, hogy elvehetünk és hozzáadhatunk az oldalhoz elemeket (oldal műszerfal, wiki, blog, naptár ...)

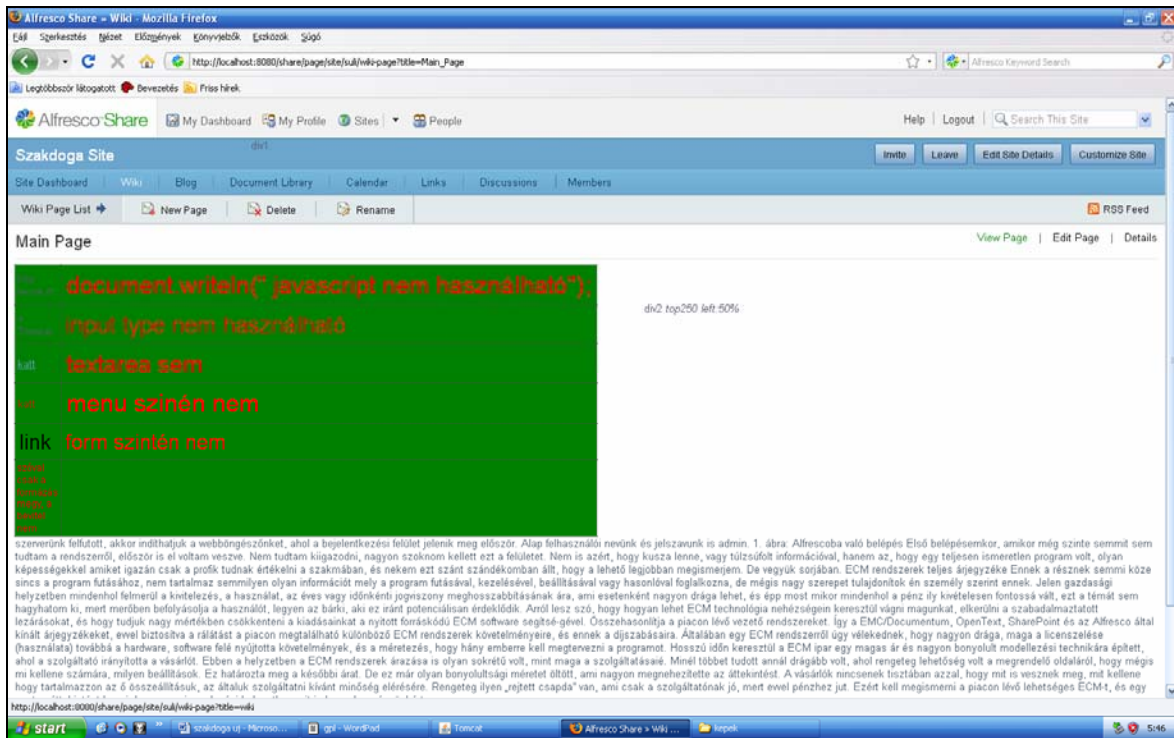
Az oldal műszerfal a következő dashlet-eket nyújtja számunkra, itt válogattam ugyanis nagy az átfedés a saját - és az oldal műszerfala között, szinte ugyan az, csak itt az oldalra vonatkozóan jelennek meg az információk. Azok ez egyedi dashletek melyek a személyi műszerfalon nem találhatóak a következők, avagy nagy az eltérés a kettő között:

- **Recently Modified Documents:** legutóbb megváltoztatott dokumentumok listája, mely felsorolja a létrehozott és változtatott dokumentumokat, melyek az utolsó hét, tizennégy vagy huszonnyolc napban lettek módosítva
- **Site Colleagues:** ez lényegében felsorolja az oldalhoz tartozó tagok névsorát és a hozzájuk rendelt szerepkört
- **Site Links:** oldal linkek dashlet linkgyűjtemény valójában, ami összegyűjti a linkeket melyek relevánsak a mi oldalunkhoz.
- **Site Wiki:** oldal wiki dashlet arra szolgál, hogy a wiki oldal komponensen megtalálható tartalmat kivonja, és megjeleníti. Lehetőség van, hogy bármelyik oldalra beállítsuk, és a nevére kattintva rögtön át is ugorhatunk rá.

Oldal elemek

- **Blog:** blog oldal komponens lehetővé teszi, hogy az oldal tartalmához kapcsolódóan hozzászólásokat, leírásokat és egyéb multimédiás megnyilvánulásokat rendelhessünk, mint kép vagy video.
- **Calendar:** kalendárium oldal komponens segítségével megfigyelhetjük az összes eseményt az oldalunkon, és az összes oldalon melynek tagja vagyunk. Eseményeket létre is hozhatunk.
- **Discussions:** vita oldal komponens arra használatos, hogy az oldal tartalmára kapcsolatos véleményeket, gondolatokat, kérdéseket, válaszokat lehessen hozzárendelni. Ez leggyakrabban kérdés-válasz formát önt, amit az emberek hozzászólásai alkotnak.
- **Document Library:** dokumentum könyvtár oldal komponens, lehetővé teszi bármely dokumentum tárolását, kezelését ami kapcsolatban van az oldallal.
- **Links:** link oldal komponens segítségével összeállíthatunk linkeket, melyek kapcsolatban vannak az oldalunk tartalmával.

- Wiki: lehetővé teszi wiki oldalak használatát. Bárki aki jogosult a hozzáféréshez az hozzájárulhat vagy módosíthat a szereplő dokumentumokon.



23. kép: Alfresco oldal wiki, css formázással részben (próba)

Share oldal elemeinek a legfontosabb sajátosságai

Dokumentum könyvtár:

- Dinamikus szűrés – olyan listát készít a dokumentumokról, melyet lehet címkék alapján, nemrég hozzáadott változtatott fájlok alapján és az alapján rendezni, hogy én vagy más változtatja e a dokumentumot.
- Flash⁴⁹ előnézet – teljesen flash alapú dokumentum nézetgető, mely magába foglalja a miniatűrök, nagyítás és teljes képernyő nézeteket, ezért semmiféle kliens oldali alkalmazást nem kell elindítani, se letölteni.

⁴⁹ <http://www.adobe.com/products/flashplayer/>

- Gazdag dokumentum adatok – név, titulus, leírás, MIME típus, címkék stb.
- Hierarchikus (fa struktúra) mappa navigáció
- Kommentálás
- Miniatűrök automatikus hozzárendelése egyes fájltypusokhoz, pl. PDF, képek, OpenOffice
- RSS hírfolyamok
- Több fájl egyszeri felvitele
- Verziófigyelés

Blog:

- Blog archívum – kereshetővé teszi a régi blogokat dátum alapján
- Blogok küldése oldalakhoz és külső blogokhoz – blogokat mint piszkozatokat küldhetünk Alfresco oldalakhoz, vagy / és külső blogos oldalakra. A szöveg írásánál van lehetőség TypePad vagy WordPress használatára, a szöveg megformázására.
- Dinamikus szűrés – olyan listát készít a dokumentumokról, melyet lehet címkék, piszkozatok, idő, külső és belső megjelenés és megjelenés dátuma szerint rendezni.
- Kommentálás
- RSS hírfolyamok

Kalendárium:

- cselekmények nyomon követése
- esemény címkézés – események címkézése és kiszűrése
- iCal – Alfresco naptárából adhatunk bejegyzéseket a gépünk naptárába
- több nézet lehetőség – nap, hét, hónap, lista alapján

Vita oldal:

- Dinamikus szűrés – címkék, új, legtöbbször aktív, összes, saját téma

- Egymásba ágyazó beszélgetések – témák és válaszaik küldése
- RSS hírfolyam

Wiki:

- Automatikus átirányítás – ha az oldal neve megváltozik, akkor automatikusan átirányítják a linket az új elnevezésű oldalra
- Automatikus cél oldal készítés – ha egy olyan linkre kattintok, ami még nem létező oldalra utal, akkor megnyitja a cél oldalt, és úgy veszi mint egy új oldal.
- Automatikus verzió és történetkezelés – frissítések automatikusan az új verzióknak fognak megfelelni
- Dinamikus oldal szűrés – címkék, nemrég megváltoztatott / hozzáadott, saját oldal és fő oldal
- Egyszerű oldal készítés
- Keresztoldal hivatkozás – egyszerű leíró nyelv használata oldalak összekapcsolására [[cél_oldal|szöveg_ami_a_link_legyen]] pl. [[wiki|katt]] → ahol a wiki oldalra hivatkozom, és a szöveg (katt) ami kioldja a linket.
- Visszatérés egy korábbi verzióra – egy régebbi verzióra tér vissza, az lesz az aktuális

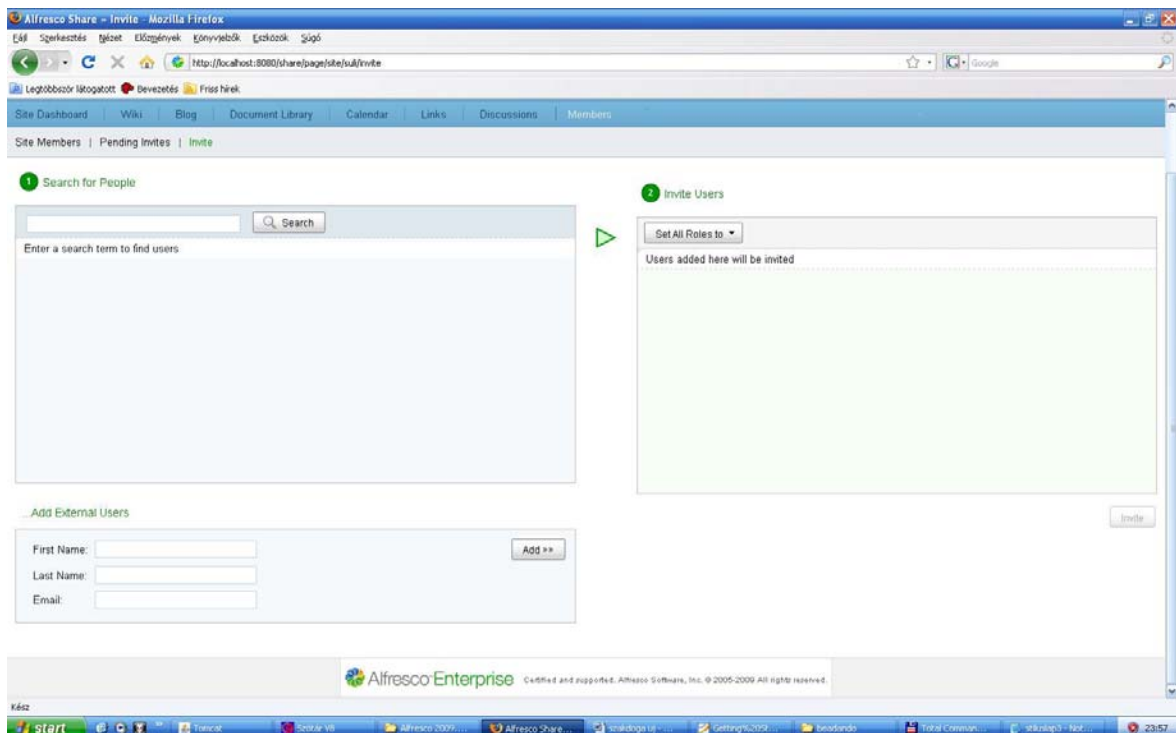
Share használata

Ha már tudjuk, hogy mire számíthatunk, milyen funkciókra, lehetőségekre a programban, akkor nézzük is meg, hogy ez miként valósul meg. Lehetőségek közül a legfontosabbakat fogom kiemelni, habár mind az, így majd próbálok válogatni, de a teljesség kedvéért a felsorolásban mindet beleteszem. Nézzük akkor.

Kiindulok abból, hogy létrehoztunk egy oldalt, az esetenben Szakdoga néven. Van egy kis leírása és nem publikus oldalként hoztam létre. Ezután egy teljesen használható úgymond közösségi portált kapunk, az összes előbb említett lehetőséggel. Itt most már csak rajtunk áll, hogy mit engedünk meg kinek, hogy nézzen ki,

milyen elrendezése legyen az oldalainknak. Evvel nem szeretnék helyet pazarolni, itt mindenki kedve szerint úgy állítja be ahogy szeretné, nagyon egyszerű lépésekkel. Amit még meg kell jegyezni, az egyszerűség kedvéért azt, hogy ha egy oldalról beszélünk itt, akkor mindenki képzeljen el egy valóságos www oldalt sok kis kapcsolódó oldallal az interneten, amihez jó esetben nem tudunk hozzányúlni, de itt igen és rögtön számos komoly tudású oldal áll a rendelkezésünkre. Ha ezzel tisztában vagyunk, akkor végre próbáljuk ki.

Mint minden közösségi oldal, ide is embereket várunk majd, hogy csatlakozni fognak, és elősegítik a munkánkat. Ide úgy tudunk csatlakozni, hogy ha az oldal tulajdonosa vagy vele egyenrangú tag meghív, avagy második lehetőségként az oldal publikus, és akkor rákeresve a join gombbal csatlakozhatunk. A meghívást az invite gombra kattintva tudjuk megvalósítani.



Két lépésből áll. Első, rákereshetünk az emberekre akik a rendszerben szerepelnek és/vagy külső emberekre, akik még nem szerepelnek az adatbázisunkban, az utóbbi esetében egy e-mail kerül elküldésre, ahol megtörténik a meghívás. Belső tagok esetén elég egy névrészlet három karaktere és máris kiválaszthatjuk őket, az add gombbal hozzá is adhatjuk a listánkhoz, akiket meghívunk. Második lépés a

kiválasztott emberek szerepkörének a meghatározása. Megadhatunk egy lépésben az összesnek egy fajtát, vagy egyenként más-más szerepkört rendelhetünk hozzájuk. Négy szerepkör közül választhatunk, ilyen a menedzser (manager), közreműködő (collaborator), munkatárs (contributor), felhasználó (consumer).

- Consumer: csak olvashat
- Contributor: Consumer összetett csoportot tartalmazza és utód csomópont létrehozására is feljogosítja (elsődleges hozzárendelés) és még egy másodlagos hozzárendelést is megenged, azaz linkkel utalhatunk a már létező csomópontokra. Ezen kívül meggátolhatja egy csomópont elérését
- Collaborator: Egy csomópont tulajdonságain változtathat, ugyanis az összes tulajdonsághoz egyetlen megkötés társul, továbbá megváltoztathatja a dokumentumok tartalmát, és meggátolhatja egy csomópont elérését és a Contributor összetett csoportot öleli fel
- Manager: embereket hívhat meg a munkába, kezeli a felhasználókat, mindenhez van jogosultsága

Engedély modellek referenciája, egy kis rálátás a jogosultságokra

Az előbb arról beszéltem, hogy egyes felhasználó típusoknak milyen jogaik, lehetőségeik vannak. Ki mire képes. Igazából ezek már nagyon összetett kategóriák, melyek bonyolult lépések eredményei. Ha belegondolunk, és kezdjük az elején, akkor van az alap négy lehetőség, amire képesek vagyunk egy fájlal kapcsolatban. Ez az olvasás, írás, törlés és a frissítés, belső berkekben CRUD, mint Create (létrehozás), Read (olvasás), Update (frissítés) és Delete (törlés). Igazat megvallva, ezek is már bonyolultabb szerkezetek, melyek pl. a Read esetében `_ReadProperties`, `_ReadChildren`, `_ReadContent`, `_ReadAssociations` és a `_ReadPermissions` alap engedélyekből állnak össze, ahol egy engedélyt „_” prefixummal és a nevével jelölnék. Ezeket tovább csoportosítva megkapjuk az összetett csoportokat, ami Read esetében: `ReadProperties`, `ReadChildren`, és `ReadContent` engedélyekből tevődik össze. Read mellett van még a Write (ami a `WriteProperties` és `WriteContent`-ből épül fel), a Delete, `AddChildren` és az `Execute` komplex engedélyek. Összetett csoportok

kombinációjából szerepköröket hozhatunk létre, mint az adminisztrátor, felhasználó, munkatárs stb. Ezeket cm:objecteknek nevezik, ahol létezik még cm:folder, cm:ownalbe és cm:lockable komplex csoport is, amelyeket a felhasználói szerepkörhöz rendelhetünk. Továbbá létezik egy Global permissions (globális engedély) néven ismert gyűjtemény, melynek lényege, hogy az összes csomópontra kifejtik érvényességüket eltekintve attól, hogy hol is vannak az adattárhelyen. A felhasználói szerepkör a Read összetett csoportot tartalmazza. A munkatárs felhasználói szerepkör, már a felhasználói szerepkört, az AddChildren (leszármazott hozzáadása) és CheckOut (csomópont lezárás a feladata) összetett osztályt is tartalmazza. Ezen felül kijelenti, hogy amit létrehoznak, azt ők fogják birtokolni, és ezt a ROLE_OWNER (globális engedély) biztosítja. Ilyen és hasonló lépésekből épül fel egy komoly jogosultságkezelő, amit manapság már XML segítségével oldanak meg.

Ez egy rövid ismertetése volt az Alfresco biztonsági előírásainak és jogosultságainak. E tanulmány keretei nem engedélyezik számomra a téma mélyebb felvázolását, így aki többet szeretne tudni a témáról, annak ajánlom figyelmébe az alábbi címeket: egyszerűbb, igazából ez az amit felvázoltam:

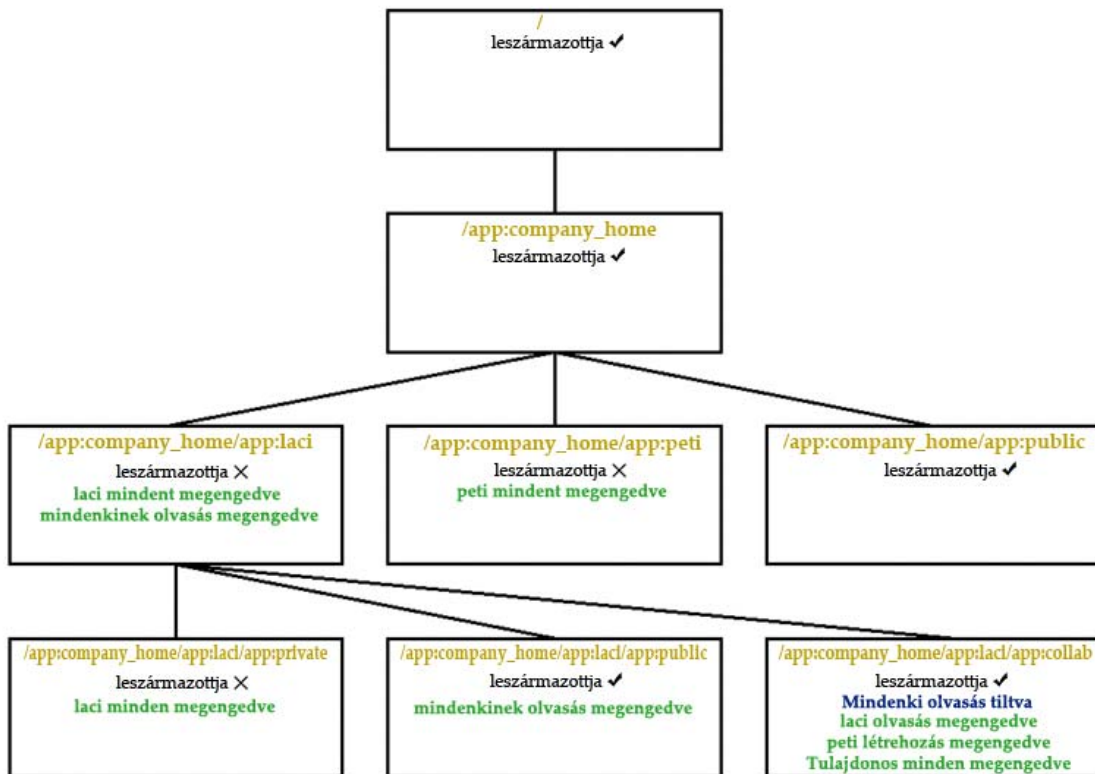
http://wiki.alfresco.com/wiki/Default_Permissions_Model_Reference

összetettebb, de átfogóbb (ajánlom):

http://wiki.alfresco.com/wiki/Security_and_Authentication

Egyszerű példa engedélyekre

Az alsó ábra egyszerű, hierarchikus engedélyeket mutat be, amelyeket egy könyvtárhoz rendeltem. Két felhasználó van, peti és laci, akikhez az engedélyeket rendeltem. Továbbá két egyedi engedélyt is tartalmaz, egyik a csomópont tulajdonosához van rendelve, másik pedig az, amelyik az összes felhasználóra érvényesül. A létrehozás, olvasás és az összes engedély-t használtam fel.



A fenti jogosultságok ábra összefoglalása:

/

Minden felhasználó olvashatja a mappa és gyermekei tulajdonságait. De viszont senki sem változtathatja meg a tulajdonságait, vagy a gyermekeiét.

/app:company_home

Szülőtől vannak örököltetve az engedélyek, és minden felhasználó olvashatja a mappa tulajdonságait, és a gyermekéit is.

/app:company_home/app:laci

Az engedélyek nem a szülő csomóponttól vannak leszármoztatva. laci minden jogosultságot megkapja. Minden felhasználó olvashatja a mappa tulajdonságait, és a gyermekéit is.

/app:company_home/app:peti

Engedélyek nem a szülő csomóponttól vannak leszármaztatva. Dave megkap minden jogosultságot, és más felhasználónak nincs semmilyen jogosultsága sem.

`/app:company_home/app:public`

Szülőtől vannak örököltetve az engedélyek, és minden felhasználó olvashatja a mappa tulajdonságait, és a gyermekéit is (inherited from /).

`/app:company_home/app:laci/app:private`

Engedélyek nem a szülő csomóponttól vannak leszármaztatva, minden jog csakis lacié és más felhasználónak nincs semmiféle jogosultsága sem.

`/app:company_home/app:laci/app:public`

Engedélyek nem a szülő csomóponttól vannak leszármaztatva és minden felhasználó olvashatja a mappa tulajdonságait, és a gyermekéit is. Ez az engedély ami erre a csomópontra érvényes elsőbbséget élvez a szülő csomóponton megadott engedély előtt, vagyis felüldefiniálja a szülő által kapott engedélyeket. Ha a szülő csomóponton egy engedélyt megváltoztatnak, az semmiféle hatással nem lesz erre az engedélyre.

`/app:company_home/app:laci/app:collab`

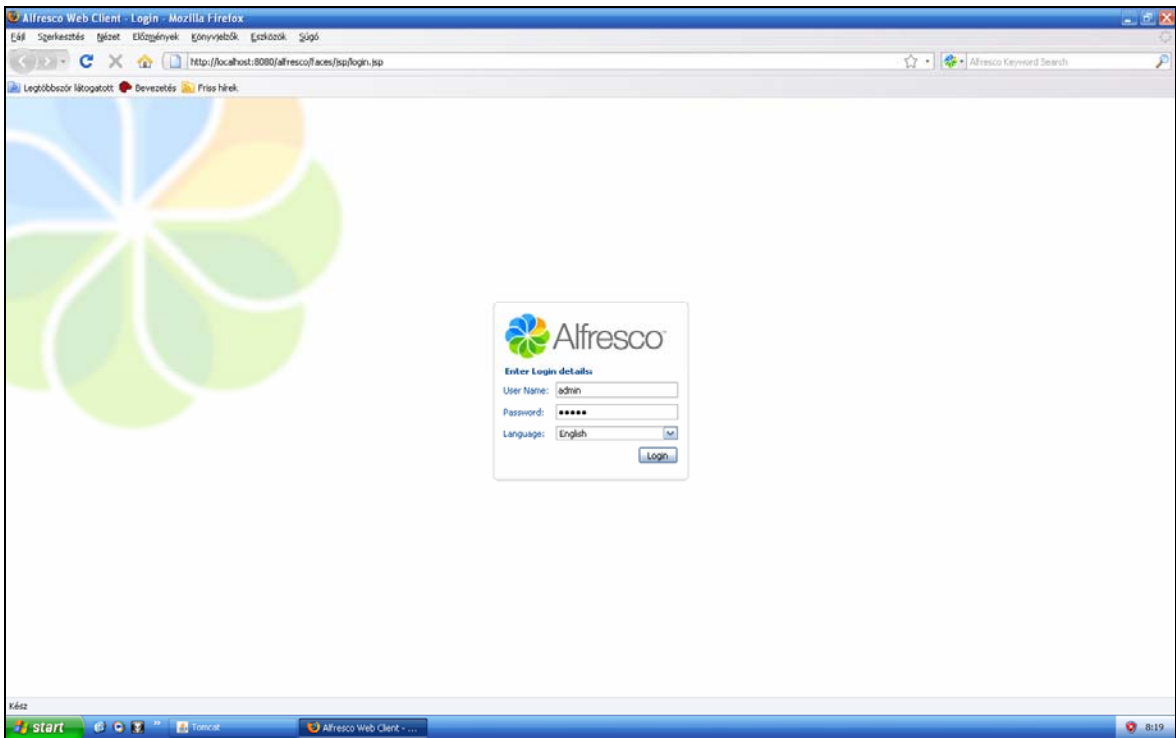
Engedélyek nem a szülő csomóponttól vannak leszármaztatva, ahol laci az összes engedélyt megkapja, peti csak a mappa és gyermekei tulajdonságát olvashatja, figyelmen kívül hagyva bármilyen szülői engedélyt. peti létrehozhat ebben a mappában gyermek mappákat, figyelmen kívül hagyva bármilyen szülői engedélyt. A csomópont tulajdonosának az összes engedélye megvan a saját mappáira. Ha peti egy fájlt készít, akkor ő rendelkezik felette, de laci elvehetné a tulajdonosi jogokat tőle, és magáévá teheti, mivel ő rendelkezik az összes engedéllyel. Ilyenkor peti már csak olvashatná a fájl tartalmát, mivel már nem ő lenne a tulajdonosa. Az összes felhasználtól visz-

szavették az olvasási engedélyt, ami felülírja a szülő csomópontban megkapott olvasási engedélyt.

Dokumentum kezelés

Ezzel rátértünk a témám legfontosabb elemére, a dokumentumkezelésre, ami magába foglalja a tartalom, internetes tartalom és a képkezelést is. Alfresco-val való ismerkedésem első lépéseiben az előbbi tény, nem volt ennyire világos számomra. Elárulom, hogy én azt hittem, úgy gondoltam, hogy minden egyes magába foglalt elemre, egy különálló programot fog szolgáltatni. De hamar rá kellett jönnöm, hogy ez nem így van, ugyanis csak egy felhasználói felületet biztosít számunkra, amelyben ötvözi az összes elem funkcionalitását. Ez az úgynevezett böngészője, amellyel a dokumentum kezelő lehetőségeinek egész tárházát elérhetjük. A dokumentumkezelőt én igazából tartalomkezelőnek hívhatnám, hisz minden dokumentum lényege a tartalom. A szerző nem dokumentumot ír le, hanem tartalmat, amit kezelni szeretnék. Ilyenkor jön be az adathordozó, egy köztes médium, amin tárolom az információt, ami lehet hagyományosan pl. papír, vagy digitálisan bármilyen formátum. Ha ez a közeg nem lenne, akkor helyénvaló lenne tartalomkezelőről beszélni, de mivel a tárolás valamilyen médiumon történik, így a jogos megnevezése a folyamatnak dokumentumkezelés. A rekordkezeléssel lehetne párhuzamosságot vonni, hisz amint egy dokumentumot elhelyeztünk egy adatbázisba, onnantól már csakis, mint rekordot emlegetjük. Olyan dokumentumkezelési szolgáltatásokra, mint amit az Alfresco nyújt csakis vállalatoknak, intézményeknek érdemes beszerezni. Ott tudja igazán megmutatni az erejét, ahol nagyon sok dokumentumot kell kezelnie életciklusoknak megfelelően, ehhez tartozó jogosultságokkal egyben, miközben a kereshetőségről, hozzáférhetőségről, küllemről és az egyszerű kezeléséről sem feledkezhetünk el.

Céges és szervezeti vonatkozásban, egy ECM rendszer használata a munkásokra nézve mindig is nehézségekkel járt. Ilyen probléma forrása többnyire az, hogy túl nehézkes volt a keresés, vagy túl komplex volt az egyes műveletek létrehozása a felhasználóknak. Ezt tükrözően manapság a cégeknél a felhasználók 5-10%-a használja az ECM rendszereket.



24. kép: dokumentum kezelőbe a belépés

Ott is főleg a háttéirodában (back office). A kifejezés még abból a korból származik, amikor a szolgáltató cégek belső elrendezése úgy nézett ki, hogy az iroda elején voltak azok az emberek, akik találkoztak a kliensekkel, és az iroda háttérében, pedig a háttér dolgozók, akiket nem láttak a kliensek, többnyire feldolgozó munkát végeztek. Ilyen felosztást találunk még a könyvtárak elrendezésében is, ahol a felhasználó az olvasószolgálatnál találkozik, de nem ismeri a háttérben zajló dokumentumfeldolgozást, rendszerezést, és egyéb folyamatokat. A cégek dolgozóinak túlnyomó többsége sajnos nem fogékony az ilyen rendszerek iránt, ami megnehezíti fejlesztésüket. Egy cégnél az elülső irodában (front office) dolgozó tudásmenedzserek használhatnak esetleg megosztott meghajtót, hogy tárolni tudják az információt, és kényelmesen elérjék azt hálón belül. Emellett az egymás közti kapcsolattartás formája az e-mail. Keresésnél, ha rendelkezésre áll is egy tartalomkezelő, akkor azt többnyire nem használják, inkább az Internetes Google keresőhöz fordulnak. Hogy e nézetét a munkásoknak megváltoztassa egy tartalomkezelő ahhoz a folyamatnak minőségbeli javulásáért, pontosságáért és megbízhatóságáért van szükség.

Azok számára, akik a saját meghajtójukon szeretnek dolgozni, és nem a megosztottnál, azoknak a dokumentumkezelő megnövelt funkcionalitást biztosít,

mint a verzióirányítás, előző verzióra való visszaállítás, akkor is, ha már a dokumentum változtatva vagy mentve lett. További funkció például a lezárás (lock), mely lezárja más elől a dokumentumot, azaz más nem nyúlhat hozzá, amíg az előző fél be nem fejezte rajta a munkákat, ha befejezte, akkor újra megnyitja utána. Igazából a dokumentumkezelés tartalomkezelésről, verzióirányításról és lezárásról szól egy megosztott meghajtón.

Dokumentumkezelőben sablonokat használunk arra, hogy a dokumentum alap tulajdonságait kinyerjük. Egy eljárás szükséges, hogy ezeket a tulajdonságokat kinyerjük pl. MS Office dokumentumból, vagy más tartalmakat, mint például képek vagy CAD⁵⁰ fájlokat. Ezekből a tulajdonságokból és meta adatokból a tartalomkezelő attribútumokat hoz létre, amit kezelni tud, ugyanis dokumentumkezelőkben a tulajdonságokat attribútumoknak hívják. E adatok kinyerése nélkül nem lenne lehetséges az adatokra való későbbi keresés. Ezen felül létezik az automatikus kivonat-készítés, ami automatikusan kinyeri a meta adatokat a dokumentumból. Ezt olyan egyszerűen teszi, mint ha magát a tartalmat emelnénk ki.

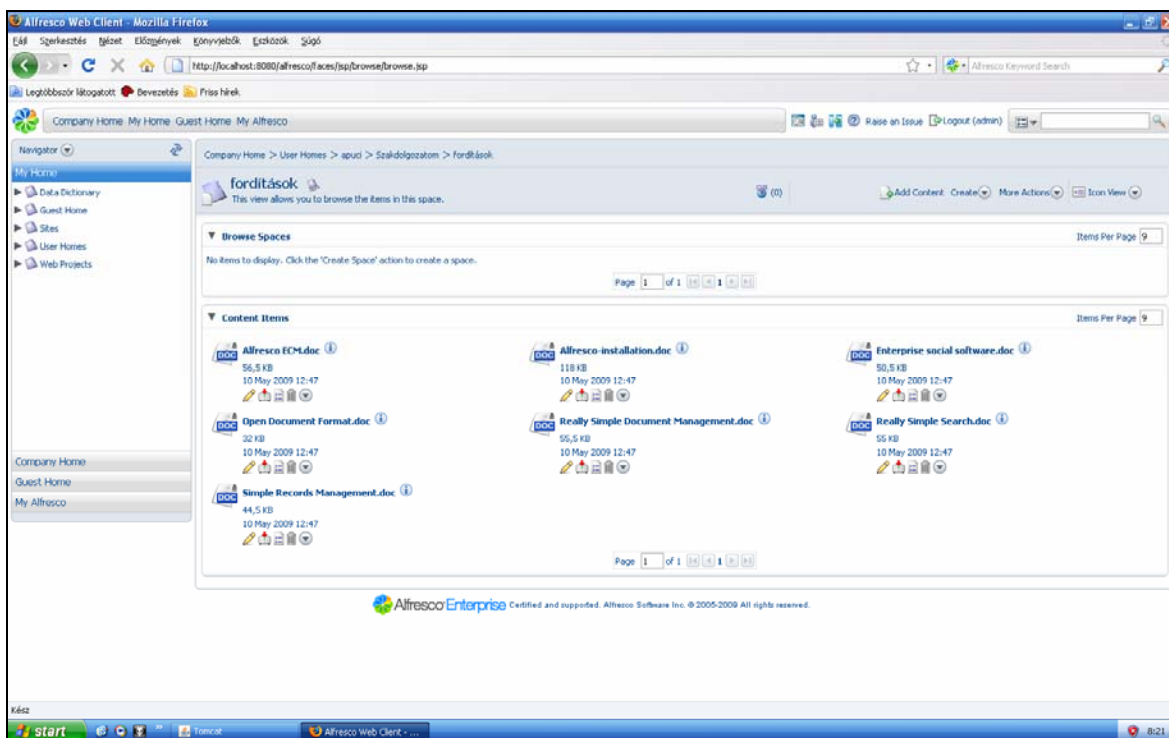
Emellett a biztonság és láthatóság is egy fontos kérdés a tartalomkezelőkben.

Biztonság

Keresésnél azon felül, hogy a kereső megkapja a megfelelő verziójú fájlt, ugyanolyan fontos az is, hogy a keresés eredményeként nem kell megkapnia az összes releváns dokumentumot. Ilyen például az, ha van egy saját dokumentumunk, és azt szeretnénk, hogy senki se lássa, és ne kereshesse meg. A biztonsági előírásnak akkor is hatnia kell, ha a keresésre releváns találatként adná meg a kereső a dokumentumunkat. Erre a megoldást a biztonsági beállítások szolgáltatják, amelyek a keresésre megszorításokat léptetnek életbe. A biztonsági előírásoknak több szintje van, legelső és legfontosabb szint az, amikor a dokumentum senki által sem látható, csak magánál a szerzőnél. Második szinten csak olvashatóvá teszi a dokumentumot. A harmadik szint megengedi a tartalomhoz való hozzájárulást (új információkkal bővíthetik a résztvevők a mappában a témát). Negyedik szinten lesz a szerkesztés engedélyezve. Utolsó szinten projekt adminisztrátorrá lehet válni, ami az 5. és leg-

⁵⁰ <http://hu.wikipedia.org/wiki/CAD>

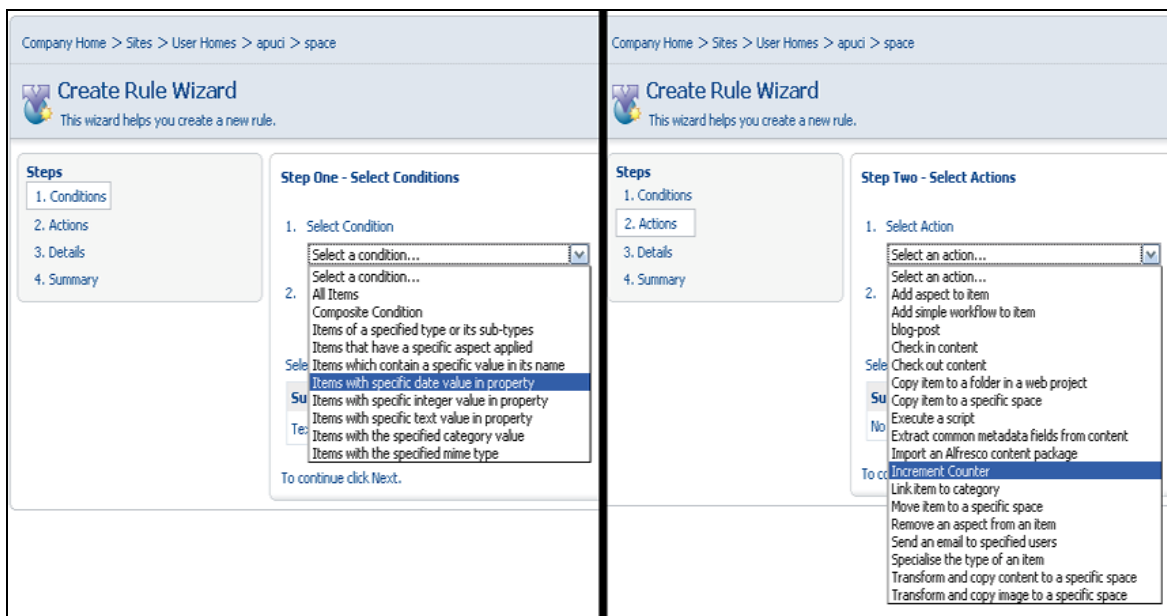
magasabb szint. Ehhez a menedzseri szerepkör lehetne hasonlítani a Share-ben. Ezen a szinten lehet az együttműködéseket kiépíteni, embereket meghívni, új szabályok és biztonsági előírásokat megadni.



25. kép: fordítások nevű space tartalma dokumentum kezelőben

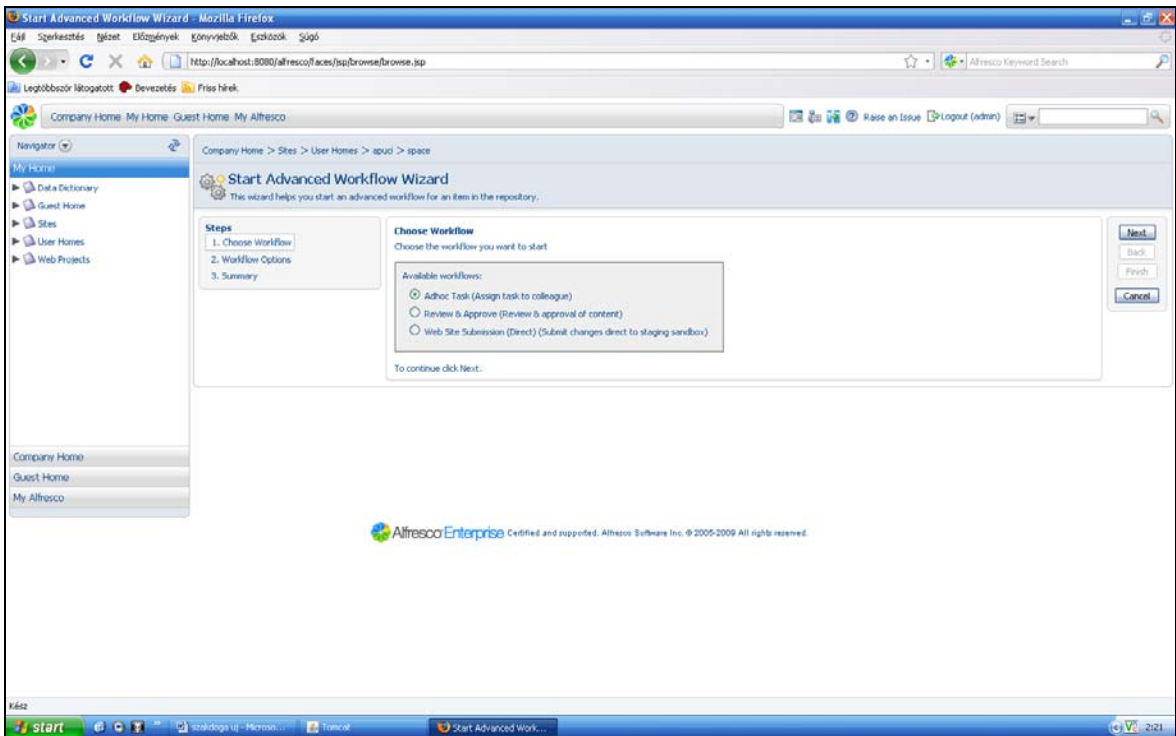
A biztonság nem statikus, időnként változik, erre jó példa a sajtóközlemény, de példának okáért, ha egy bank gazdasági céllal hozott létre dokumentumot, akkor az valószínűleg privát lesz egy magas biztonsági szinttel ellátva. Ezután az átnézés fázisába kerül, ahol csakis a bírálók (reviewers) láthatják. Ha erre megfelelt, akkor mehet csak tovább a következő szintre, ahol egy másik különböző biztonsági ellenőrzés hatálya alá kerül, ami megengedi a hozzászólások és vélemények kinyilvánítását belső munkatársak számára. Ha itt is megfelelt, akkor csak is akkor léphet a következő biztonsági szintre, ahol már publikus dokumentummá válhat, ha azt szeretnék. Egy tartalomkezelő rendszerben, az összes ilyen lépés egyszerűen le van kezelve életciklussal. Ha a dokumentum egyik állapotból a másikba jut, és legvégén az space-be (igazából mappa csak többet tud nála), akkor a biztonsági szintje, ami eddig hozzá volt rendelve az automatikusan az új állapotnak megfelelően fog megváltozni. Ehhez hasonlóan, ha egy space státusza megváltozik, a benne lévő fájlok státusza is vele együtt fog. Az életciklus továbbá automatikusan feljegyzi a változta-

tásokat, azt hogy ki által volt megírva és mikor, ki bírálta és mikor, milyen vélemények születtek, továbbá az, hogy mikor volt jóváhagyva és mikor lett nyilvánossá nyilvánítva. Erről egy log fájlt kapunk. Az életciklus sok különböző fázist és folyamatot felel, és bonyolultságát leplezve mégis olyan egyszerűen lehet létrehozni.

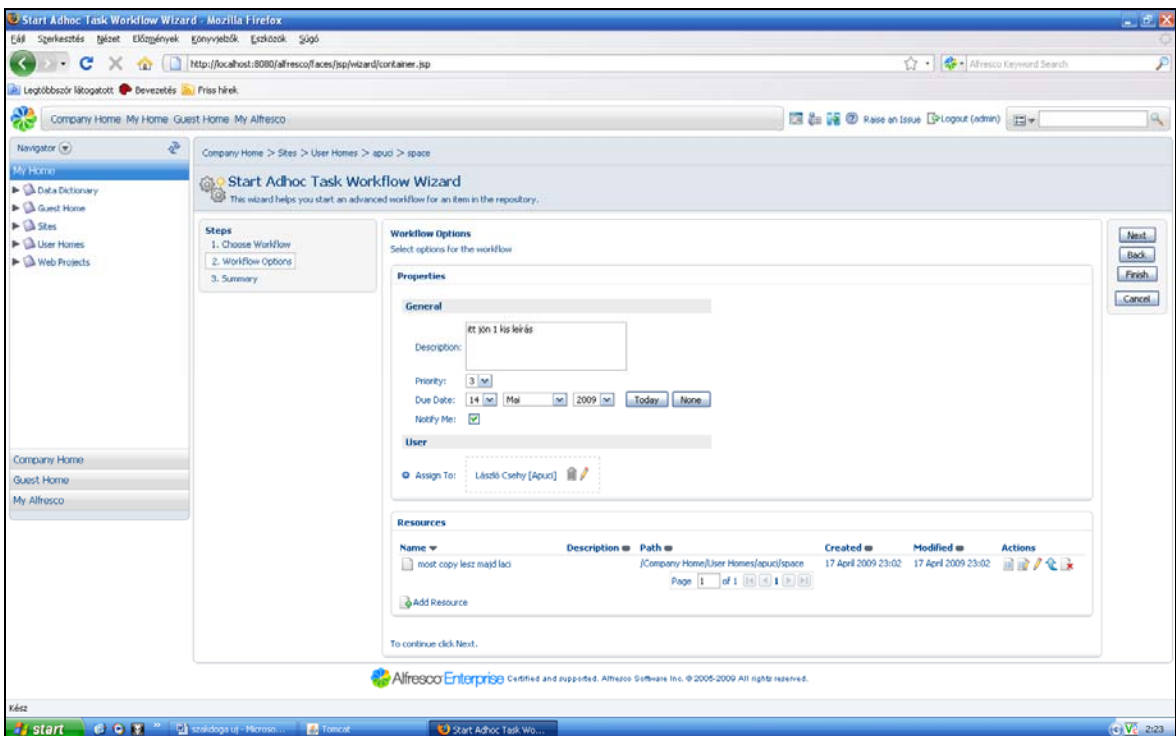


26. kép: szabályok létrehozása, bal oldalt a feltétel, jobb oldalt a reakció

Ha egy dokumentum egy tökéletes életciklust produkált, akkor ha például MS Word formátumban volt közzétéve, akkor utólagosan könnyen megváltoztatható, ahhoz hogy ezt megakadályozzuk, kibocsátáskor csak olvasható formátumban, például PDF lesz közzétéve a dokumentum kezelő rendszerből. Egy fejlettebb lehetőség a dokumentum kezelés végrehajtására, az hogy a vízjeleket vagy digitális jog kezelést adunk a kibocsátáskor a dokumentumra. Erre példa az, hogy egy indítványt kiadhatunk, ami 30 nap után lejár. Igazából arról szól, hogy a kibocsátani kívánt dokumentumot egy olyan formátumban adjunk közre, ami csak olvasható és nem olyanban, ahol a biztonság hiányozhat, azaz megváltoztatható. Ha ezeket a tulajdonságokat egyesítem, akkor dokumentum modellt kapok. Minden dokumentum egy mappában él, és minden mappának vannak saját tulajdonságai. Minden dokumentumhoz tartozhat hozzászólás, lehet más formátumban, lehet fordítása a dokumentumnak, ezt rendition-nek hívjuk, például eredeti magyar MS Word formátumú fordítás szlovák PDF formátum.



27. kép: mélyebb munkamenet készítés 1. lépés



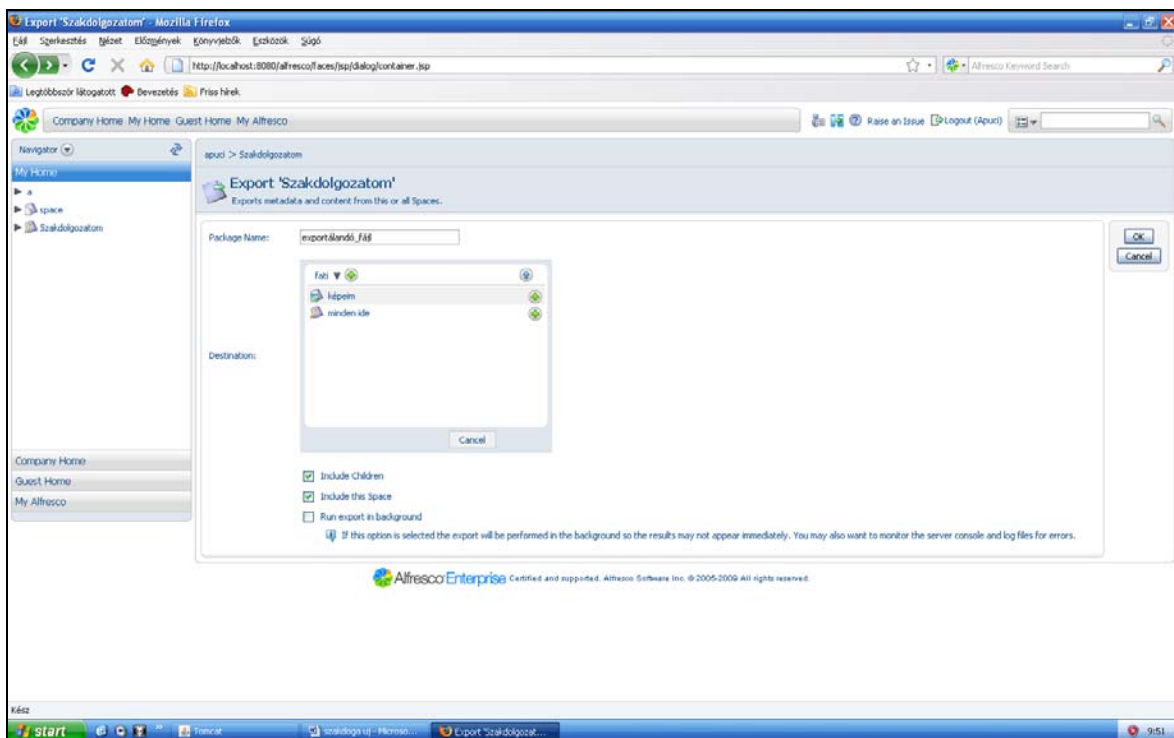
28. kép: mélyebb munkamenet készítés 2. lépés

Minden dokumentum rendelkezik verzió kezeléssel és naplózással. Ez egy alap dokumentum-modell, ami kezelésre szorul egy dokumentumkezelő rendszerben. Fejlettebb dokumentumkezelő ezeket a funkciókat öleli fel.

- Életciklus felöleli az irányítást, helymeghatározást, biztonságot, formátumot, fordításokat és kibocsátásokat.
- Nyelvi fordítások (language renditions): ha valaki német operációs rendszert használ, akkor ez felismerődik, akkor az adott nyelv nyelvén publikált dokumentumok jelennek meg, mint találati halmaz, vagyis a tartalom van nyelvileg lekezelve.
- Rekord kezelés és megrekedés: ez biztosítja azt, hogyha egy dokumentum lejárt az akkor sincs törölve egy időn keresztül, amíg törvényes okok miatt szükséges.

Alfresco dokumentumkezelése

- Használhatóság kényelme: arról szól, hogy olyan koncepciókat használjunk, amiket már ismerünk. Az emberek ismerik a megosztott meghajtó fogalmát, ezért az Alfresco adattárhelyét úgy alakította ki, mint hogyha egy megosztott meghajtó lenne. Mappákhoz (spaces) is könnyen lehet szabályokat rendelni. Az Alfresco-t olyan könnyű keresni mint a Google-t. Lehet kategóriákban keresni, mint a Yahoo-ban.
- Megosztott meghajtói felület: az Alfresco adattárhelyének van egy egyszerű és eredeti CIFS-je (közös internet fájlrendszer, Common Internet File System) és az erre épülő megosztott meghajtói felülete, ami tisztán java-ban íródott. Ez megengedi azt hogy a felhasználók a szó szoros értelmében, például a C meghajtójukra menjenek és egyszerűen vigyenek be elemeket az Alfresco adattárterületére. Sőt az Alfresco-ból még dokumentumokat is kivethetnek egyenesen bele a táska szolgáltatásokba (brifcase), ugyanis a Windows azt hiszi, hogy egy megosztott meghajtót ér el vele.
- Munkafolyam: olyan munkafolyamot hoztak létre, amely logikusan végigkövethető, és tükrözi a csapatok reális munkafolyamát. Gyakorlatban úgy oldották meg, hogy a mappán sok név szerepel és aki már olvasta, az csak egyszerűen kihúzza a nevét. Ezt tükrözték le az Alfrescoban. (27. és 28. kép)
- Smart Space (okos terület): Egy termék indításkor vagy ismétlődő projektek-nél egy szervezetnek sok mappája és sok sablon dokumentuma lesz, amire



29. kép: okos terület exportálása

- Tulajdonság és meta adat kiemelhetjük

Ha egy word dokumentumot beemelünk az Alfrescoba, akkor az automatikusan kiemeli a tulajdonságokat és a meta adatokat és a dokumentumkezelő rendszer által használható attribútumokká alakítja. Itt lehetséges a tartalom tulajdonosa kulcsszavak, dokumentumcímek és egyéb információk automatikus beírása a rendszerbe gépelés nélkül. Ezzel megalapozva a jövőbeli kereshetőséget.

- True Zero Footprint Client: az Alfresco hiszi, hogy a használhatóság kényelmébe beletartozik az alkalmazások kiadásának könnyítése is, ezért az Alfresco programoknál a kliens gépen semmit sem kell telepíteni, és mégis mindent használhatunk. Ezt hívják True Zero Footprint Client-nek. Ezért

Egyszerű munkafolyamatra a mappát használja. Így ha egy dokumentum bekerül egy mappába, akkor meg lehet szabni, hogy automatikusan egy más formátumba például PDF, ODF formátumra legyen alakítva, és még sok egyéb is meghatározható. (26. kép). Minden egyes mappának külön biztonsági szintjei vannak.

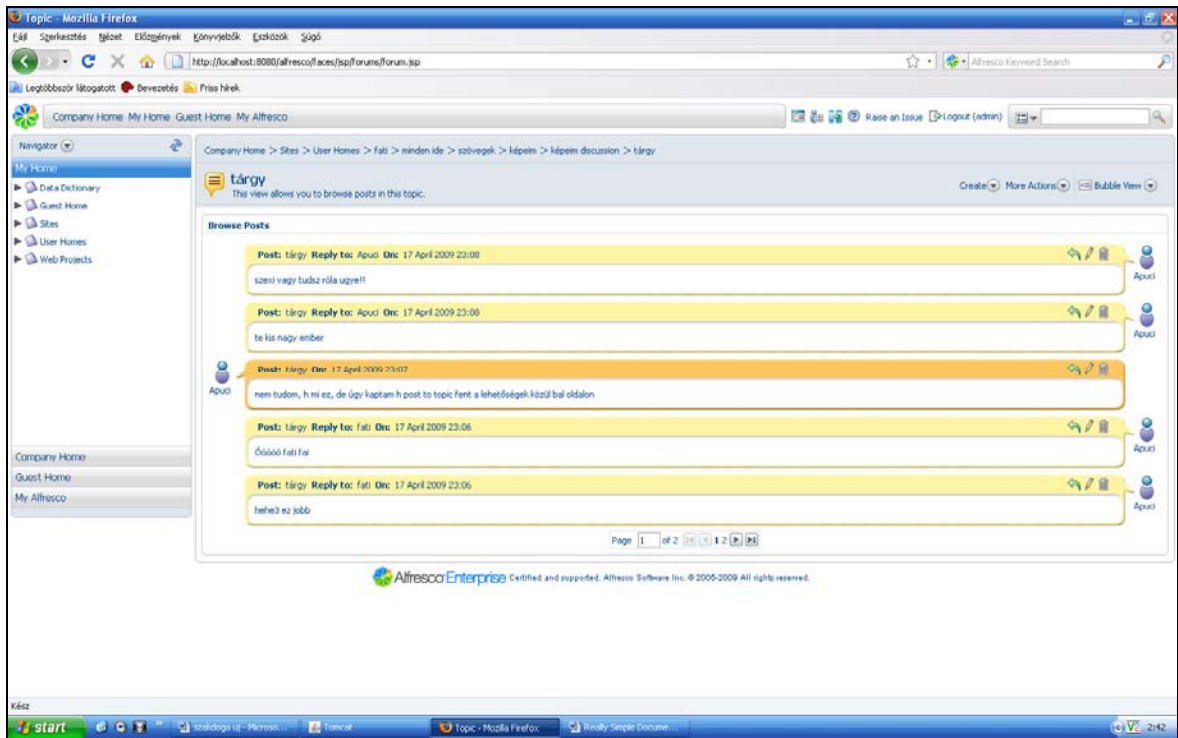
MS Office termékhez van beépülő modulja, ennek lehetőségei melyeket a felhasználóknak megad:

- be és kijelentkezéseket
- hogy Alfresco adattárhelyeket böngésszenek egy új panelen keresztül
- kereshessenek
- két különböző verziójú dokumentumot összehasonlítsanak
- munkafolyamokat ütemezhessenek naptár alapján
- új dokumentumokat hozhassanak létre
- verziófigyelést

Eddig ezek a lehetőségek csakis a dokumentumkezelőben voltak, de ezáltal az MS Office-ban is elérhetővé vált, ami egy csomagként hozzárendelhető az Office-hoz.

Továbbá beszélgetéseket lehet engedélyezni, ilyen üzenetküldések formájában. Ezt a fórum engedélyezi, amit minden egyes dokumentumhoz, space-hez hozzá lehet rendelni, és azok írhatnak ide, akik jogosultak, vagy bárki ha publikus (30. kép).

Ezenfelül egyszerű HTML és txt dokumentumokat online is szerkeszthetünk, egyenesen a dokumentumkezelőben. Vagy létrehozhatunk egyszerű txt, HTML-t és XML fájlt is (31.kép.)



30. kép: fórum külalakja a dokumentumkezelőben

Összegzés

Munkám a vállalati tartalomkezelésről szólt, ami nagyon fontos szerepet tölt be egy vállalat életében. Nem egy egyszerű programról van szó, nagyon összetett és komoly munkát jelentett a megismerése számomra. Attól a pillanattól kezdve, amikor elkezdtem ismerkedni a programmal, tudatosult bennem, hogy nagy kihívások elé nézek. Programom dokumentációja és leírásai csakis angol nyelven léteznek, és így nagyon sok fordítást kellett végezni, ami megnehezítette az értelmezést és a gyors haladást. Ezért legelőször is a fordításnak kezdtem neki, majd az összegyűlt anyagot kiválogattam és belevágtam munkám megírásába. Folyamatosan ahogy haladtam a témákon, lehetőségeken keresztül, ráeszméltem, hogy az addigi fordításaim és erőfeszítéseim egy nagyon komoly téma felé sodortak. Annyi lehetőség, annyi opció, és annyira összetett, komoly dolgok ezek, hogy féltem munkám sikerének minőségének kimenetelétől. Ha visszagondolok az elmúlt fél évemre, és szenvedéseimre amit e munka megírása okozott nekem, azt mondom komoly. De számomra megérte, mert nagyon sok hasznos információval, remélem tudással is gazdagodtam, amit ha legalább egy kicsit is átadhatok másnak, az elégedettséggel töltene el.

Köszönöm.

Képjegyzék

31. kép: telepítés során 4x kaptam 2 különböző gépen ezt a hibát
32. kép: Documentum, OpenText és Share Point árazásának összehasonlítása Alfresco-éval
<http://share.alfresco.com/share/proxy/alfresco-feed/api/node/content/workspace/SpacesStore/cf03ed03-00c2-4a0b-bf8c-59910781c811/Alfresco%20White%20Paper%20-%20TCO%20for%20ECM.pdf>
14. oldal, 2009.04.02.
33. kép: ECM rendszerek összetett árazása három aspektus alapján
<http://share.alfresco.com/share/proxy/alfresco-feed/api/node/content/workspace/SpacesStore/cf03ed03-00c2-4a0b-bf8c-59910781c811/Alfresco%20White%20Paper%20-%20TCO%20for%20ECM.pdf>
15. oldal, 2009.04.02.
34. kép: login.jsp licensz részlete, ami a C:\Alfresco\tomcat\webapps\alfresco\jsp könyvtárban található
35. kép: Alfresco egyszerű felépítése
http://wiki.alfresco.com/wiki/Image:Alfresco_Repository_Architecture_diagram1_colored.png
36. kép: Alfresco Spring Framework architektúrája, és elemek
http://wiki.alfresco.com/wiki/Image:Alfresco_Repository_Architecture_diagram2_colored.png
37. kép: szolgáltatások és kommunikációs protokollok, majd azok végső szolgáltatásai
http://wiki.alfresco.com/wiki/Image:Alfresco_Repository_Architecture_diagram2_colored.png
38. kép: bal oldalt egy egyszerű XML leírás, míg jobb oldalt a megfelelője IE böngészőben
39. kép: egy egyszerű külső css formázás hozzáadása az XML dokumentumhoz
40. kép: a példa „nyers” XML megjelenítése
41. kép: XSL stíluslap használata XML dokumentumon
42. kép: Alfresco által használt MIME típusok listája
43. kép: összetett és egyszerű keresés
44. kép: Alfresco adattárhely hozzáadás
45. kép: adattárhely hozzáadva
46. kép: összetett keresés eredménye és keresés mentése
47. kép: böngésző keresés eredménye
48. kép: iwiw felhasználói kapcsolat meghatározása
49. kép: belépés saját gépemen az Alfresco közösségi oldalra
50. kép: internetes belépés az Alfresco Share dokumentum könyvtárába
<http://share.alfresco.com/share/page/site/community/documentlibrary> 2009.05.04.
51. kép: Alfresco Share műszerfala, jól kivehetők a Dashlet-ek
52. kép: Alfresco Share internetes műszerfala
<http://share.alfresco.com/share/page/site/community/dashboard> 2009.05.01.
53. kép: Alfresco oldal wiki, css formázással részben (próba)
54. kép: dokumentum kezelőbe a belépés
55. kép: fordítások nevű space tartalma dokumentum kezelőben
56. kép: szabályok létrehozása, bal oldalt a feltétel, jobb oldalt a reakció
57. kép: mélyebb munkamenet készítés 1. lépés
58. kép: mélyebb munkamenet készítés 2. lépés
59. kép: okos terület exportálása
60. kép: fórum külalakja a dokumentumkezelőben

Hivatkozásjegyzék:

1. <http://www.alfresco.com/> 2009.05.11.
2. http://wiki.alfresco.com/wiki/Main_Page 2009.05.11.
3. <http://forums.alfresco.com/en/> 2009.05.11.
4. <http://forge.alfresco.com/> 2009.05.11.
5. <http://blogs.alfresco.com/planet/> 2009.05.11.
6. <http://blogs.alfresco.com/wp/opentalk/> 2009.05.11.
7. <http://www.w3c.org/> 2009.04.28.
8. <http://en.wikipedai.com/> 2009.05.01.

Köszönetnyilvánítás

Dolgozatom lezárás ként szeretnék köszönetet nyilvánítani oktatómnak és témavezetőmnek Dr. habil. Boda Istvánnak, hogy segítette munkámat.

Dr. Kovács Zsigmondnak akinek a témaválasztásomat köszönhetem.

Szüleimnek akik tanulmányi éveimben mindenben segítettek, és nélkülük most nem tartanék itt.

Külön köszönetet szeretnék adni Jadallah Fatimének, aki segítette munkámat, és mellettem állt e nehéz időben.

Köszönöm mindenkinek.