

ÜGYVITELI ALKALMAZÁSOK MEGVALÓSÍTÁSA WEB-ALAPON

Kelemen Balázs, balazs.kelemen@arvato-systems.hu

arvato systems Hungary Kft.

1 Bevezetés

Az iparban már évek óta elfogadott gyakorlat, hogy komolyabb ügyviteli funkcionalitást megvalósító alkalmazásokat is vékonykliens architektúrával valósítanak meg. A vékonykliens előnyei ma már közismertek: a kliens gépekre nem történik szoftver specifikus telepítés, központosított az adatok, a folyamatok és az erőforrások kezelése. Már komolyabb ergonómiai elvárások esetén is inkább hibrid megoldásokat javasolnak. A vastagkliens architektúra kizárólag erős grafikai funkcionalitás esetén jöhet szóba, vagy kliens oldali perifériák kezelésekor. Különösen érvényesek ezek a megállapítások manapság, amikor a web2-nek is nevezett technológia megveti a lábát a vékonykliens rendszerekben. Ez a technikai megoldás kliens oldalon valósít meg magas fokú ergonómiai funkcionalitásokat, ezzel mintegy a hibrid megoldások közé emeli az eddig klasszikusan vékonykliens alkalmazásként emlegetett web-es rendszereket.

Szerencsére egyre több szervezet ismeri fel az egyedi (vagy testre szabott) ügyviteli megoldás hatékonyságát. Ezen rendszerek azonban nem kezelik rugalmasan a modern kor legújabb igényét, a kollaboratív, elosztott munkavégzést.

A tökéletes megoldást az összes adat strukturált tárolása jelentené (és tulajdonképpen ez az ügyviteli rendszerek alapja), mégis főleg kollaborációs munkalépésekben azt tapasztaljuk, hogy a szervezetek előszeretettel fordulnak strukturálhatatlan megoldásokhoz, például személyes email kommunikációt alkalmaznak, esetleg Microsoft Word formátumban adják át egymásnak az anyagaikat, vagy jobb esetben strukturált Microsoft Excel állományokon keresztül kommunikálnak. A Microsoft Word rendelkezik korrektúra funkcióval, de tapasztalatok azt mutatják, hogy ennek használhatósága erősen korlátozott. Megoldást jelent egyes esetekben a wiki technológia, de ez kizárólag dokumentatív kollaborációra használható, és ott is problémás a jogosultság kezelés.

Jelen tanulmány apropója egy 2005-2006 évben GVOP keretében elkészült projektmenedzsment rendszer [1]. A rendszer készítésekor szembesültünk a fent felvetett problémákkal és próbáltunk rá megoldást találni.

2 Ügyviteli alkalmazás

Ügyviteli alkalmazás alatt tipikusan egy cég (szervezet) egyedi folyamataira, adataira tervezett szoftverrendszert értünk. Egyes esetekben ügyvitelre alkalmazunk általános

tervezésű szoftverrendszert, illetve ezeket az általános célú rendszereket szabhatjuk testre a szervezetnek megfelelően.

Az ügyviteli rendszerekben adatstruktúrák realizálódnak, amik relációs adatbázist képeznek. Ezek a struktúrák hűen tükrözik a szervezet folyamataiban megjelenő adatokat és adatkapcsolatokat.

A szervezet folyamatainak lebonyolítását folyamatvezérlő, illetve maga az üzleti logika valósítja meg.

2.1 Dokumentum kezelés

Egyes folyamatlépéseken, illetve kollaborációt (ügyfélkapcsolatot) igénylő feladatoknál óhatatlan, hogy dokumentum jellegű anyagok állnak elő. Ennek már csak azért is jelentősége lehet, hiszen egyes anyagokat jogszabályi kötelezettségek alapján irattárban kell iktatni, tárolni. Például határozatok, vagy válaszlevelek.

Ideális megoldást az jelentene, hogy bár a szoftver riportszerűen képes előállítani dokumentum típusú kimenetet, de hasonló anyagok értelmezését nem végzi.

Sajnos azonban az tapasztalható, hogy a már meglévő ügyviteli szoftverek hiányosságait pótlandó teljesen indokoltan Office alapú Word, illetve Excel nyilvántartások készülnek. Nem nehéz belátni, hogy egy ilyen ad hoc nyilvántartás utólagos üzleti felhasználása még alapos migráció után sem garantált. A kötetlen, strukturálatlan adattartalom okozta következményeket, nagy mértékben súlyosbítja a Microsoft alapú rendszerek nyíltságának hiánya. Ebből adódik, hogy az ilyen formátumokat kezelő nyílt felhasználású rendszerek mind a mai napig gyerek cipőben járnak. Például a [POI](#) [2] projekt.

A kollaboráció eszközül a szervezetek előszeretettel választják a korrektúrával ellátott Word formátumot. Illetve az ilyen dokumentumok továbbítását a felek között. Az elmélet lényege a változások követhetősége. Sajnos tapasztalat alapján elmondható, hogy ennek a funkcionalitásnak a használata jelentősen korlátozott, illetve üzemi környezetben nem alkalmazható. Megoldást jelenthet egyfajta wiki jellegű kollaboráció, de ennek hátrányai közé azonban olyan jelentős érvek sorakoznak, mint az internetkapcsolat-mentes szerkesztés hiánya, a problémás jogosultságkezelés és a felhasználói ellenállás.

Ügyfélkapcsolat bonyolításához a kifogástalan külalak elengedhetetlen és ezt a követelményt a Microsoft Word százszázalékosan kielégíti. Az ügyviteli szoftverek képesek sablonok alapján saját adat-rendszerükből ilyen típusú formátumot előállítani. Ha pedig a kimeneti formátumot PDF formátumra állítjuk, akkor külalak megtartása mellett a dokumentum védelmet is megkaphatjuk. Itt kell megemlítenem, hogy bár mind Microsoft Word, mind PDF formátumok ingyenes szoftver segítségével megtekinthetők, a szerkesztésükhöz használt Microsoft Office szoftvercsomag licencdíja nem elhanyagolható.

Mindkét problémára megoldást fog nyújtani a nyílt szabványú [Open Office XML](#) [3] formátum, ennek bevezetésére azonban még éveket kell várni.

Bár a Microsoft Excel formátumban érkező anyagok rendszerbe töltése elvileg megoldható, a kötetlen hozzáférés miatt helyette OnLine úrlap alkalmazása javasolt. A Microsoft Word formátumban érkező anyagok földolgozása a probléma jellegéből adódóan nem kivitelezhető, azonban az ilyen anyagok analízálhatók például tartalmuk kereshető. Példa erre a [Lucene projekt](#) [4] megoldása.

További problémát jelent, hogy bár a felhasználók előszeretettel szerkesztik az anyagaikat Word szerkesztővel, nyilvánosságra hozásukat mégis PDF formátumban szeretnék elérni. Word formátum átalakítása PDF formátumba még a piacon kapható eszközökkel sem triviális,

nem hogy ingyenes megoldásokkal. Az automatikus átalakításra pedig egész egyszerűen nincs megbízhatóan működő megoldás. A cégünk által ajánlott megoldás erre a problémára egybevág egyes ingyenes átalakító szoftverek működésével: a virtuális nyomtató-meghajtóként települő PDF meghajtóra automatikus nyomtatást küld a rendszer, majd az elkészített PDF állományt csatolja az eredeti Word formátumú anyag mellé.

3 "Promóció" projekt

A "Promóció" nevű projekt egy 2005-2006 évben GVOP keretében elkészült projekt-menedzsment rendszer. A rendszer főbb funkciói:

- Projekt tervezés
- Pénzügyi tervezés
- Teljesítés könyvelésének lebonyolítása
- Projekt monitorozás
- Automatikus dokumentálás (Például indikátorok követése GVOP-s projekthez)
- Résztvevők közti kollaboráció
- Multimodalitás

Vagyis a rendszer képes egy projekt teljes életciklusának lebonyolítására a konzorciumban lévő partnerek tevékenységeinek összehangolására.

3.1 Architektúra

A Promóció rendszer teljes egészében Web alapokra lett építve. Úgy lett megtervezve, hogy a felhasználókat képező egyes konzorciumi tagok lokálisan is szeparálva tudjanak együtt dolgozni: a rendszert mindenki interneten keresztül éri el.

A web alapú architektúra elérésére portál funkcionalitást szervesen magában hordozó keretrendszer került kifejlesztésre Java programnyelven.

A keretrendszer bázismoduljai közé tartozik az adat-perzisztencia kezelését végző modul, és az alapvető portál funkcionalitások, mint például a portletek kezelése vagy a beléptetés.

A keretrendszer atomi szinten támogatja a modulok kezelését, tulajdonképpen minden funkcionalitás modulként kerül betöltésre a keretrendszerbe.

3.2 Keretrendszer

Ahogy az előző fejezetben említésre került a portál motor egy általános funkciójú, moduláris felépítést támogató keretrendszerbe épül be.

3.2.1 Adatbázis elérő modul

A keretrendszer fontos tulajdonsága, hogy biztosítja az adatelérést a további modulok számára. Az adatelérő modul perzisztens adathozzáférést biztosít. Ennek az a lényege, hogy magasabb szinten lévő logikának nem kell az adatbázis kapcsolat kezelésével törődni, ezek a rutinok az adatbázis adatokat már struktúraként használják. Olyannyira sikerült ezt a funkcionalitást elérnünk, hogy az adatbáziskezelő típusának cseréje nem jár további kódolási tevékenységgel.

A keretrendszer adatbázis elérő rétege rendkívüli dinamizmussal lett felruházva. Például egyes üzleti objektumok futás közben is kiegészíthetők tulajdonságokkal, ezek a

tulajdonságok menthetők adatbázisba. Ez azt fogja jelenteni, hogy az objektumot tároló tábla futás közben kibővül egy oszloppal. Az adatelérő réteg azt a filozófiát követi, hogy minél egyszerűbben állunk hozzá a problémához, később annál jobban testreszabható rendszert tudunk belőle összeállítani.

Az adatbázis elérő réteg többnyelvűsége lett megtervezve, ebből is látszik, hogy már tervezéskor figyeltünk a későbbi portál funkcionalitásra. A nyelvi támogatás olyan kifinomult, hogy például a rendszerben megfogalmazható a következő kérdés: *“Keresem azokat az GazdaságiÁgazat típusú objektumokat, ahol a hivatkozott Jogszályban szövegében szerepel a “kezes” szó, és az ágazat leírásának angol nyelvű szövegében szerepel a “van” vagy a magyar szövegben “must” kifejezés.”*

Mint látható a rendszer komoly kereső-szűrő funkcionalitással bír és képes kezelni a hivatkozásokat egyes objektumok között. Objektumok keresésekor az erre szolgáló saját fejlesztésű API-val kritériumokat állíthatunk elő logikai összefüggésekből, vagy komplikáltabb lekérdezés esetén közvetlen SQL utasítással is próbálkozhatunk. A keresés eredménye mindig egy objektum halmaz lesz, amit nagy találati lista esetén ablakozni célszerű (ha ezt az adatbázis kezelő támogatja).

3.2.2 Portál motor

A portál funkcionalitás szinkronba lett hozva a keretrendszer nyújtotta lehetőségekkel, így a perzisztencia réteggel karöltve egyszerűen lehet a rendszerben adatelérő képernyőket készíteni. Sőt a keretrendszer részeként olyan eszközhalmaz lett kifejlesztve, amivel adatstruktúra-leírók alapján testreszabható portál generálható egy gombnyomásra.

A portál megvalósítja a kor elvárásainak megfelelően az MVC modellt, vagyis a felhasználó által kezdeményezett események lefolyásába az összes érintett megjelenítési elem beleszólhat.

A portál a fent említett többnyelvűsége alapozva támogatja a feliratokat. Ennek megfelelően a felhasználói felületen megjelenő minden felirat (szoftver szöveg) az adatbázisban kerül eltárolásra akár több nyelven is. Így a feliratok utólagos javítása, teljes rendszer átszövegezése újabb nyelvre nem igényel semmilyen programozói beavatkozást.

A portál rendszer természetesen támogatja a protlet struktúrát, ezeken belül, pedig ajánlást ad tulajdonság oldalak, listázó-szűrő oldalak elkészítésére, mégpedig úgy, hogy generálja azokat. A portál természetesen megoldást nyújt az alapvető portálkövetelményekre úgy mint a tartalomkezelés és a navigáció.

A portál egyes elemeinek megjelenítési rétegéhez a Velocity [5] eszközt hívtuk segítségül. Úgy tapasztaltuk, hogy ezzel a megoldással megfelelő hatékonysággal lehet html formátumú dinamikus kimenetet előállítani.

3.2.3 Modulok és technikai megfontolások

Mint ahogy az említésre került a keretrendszer minden további bővítése modul kapcsolásával valósítható meg. Ez nem csak a hatékony újrafelhasználhatóság szempontjából fontos, hanem egy modulokból (konzerv megoldásokból) összeállítható, testreszabható rendszer elkészítése lényegesen kisebb időt vesz igénybe, és ennek stratégiai jelentősége van.

A kifejlesztett keretrendszerbe szervesen illeszkedik a már említett adatbázis elérő (perzisztencia) modul is. A jogosultság kezelő modul képes a felhasználókat csoportokba rendezni a szervezeti hierarchia szerint, illetve a rendszerben végezhető tevékenységeket

házi rendekbe szedni az üzleti modell szerint. A házi rendek és a felhasználói csoportok illetve felhasználók összekapcsolása valósítja meg a jogosultság kezelést.

A fejlesztés során természetesen számos további modul került megvalósításra, ezekből a cikk témájának megfelelően a Dokumentumkezelő modul kerül tárgyalásra a Dokumentum menedzsment fejezetben.

Gyakran merül föl a kérdés külső szemlélőkben, hogy vajon miért nem használunk nyílt forráskódú megoldásokat perzisztencia, illetve portál megoldásokra.

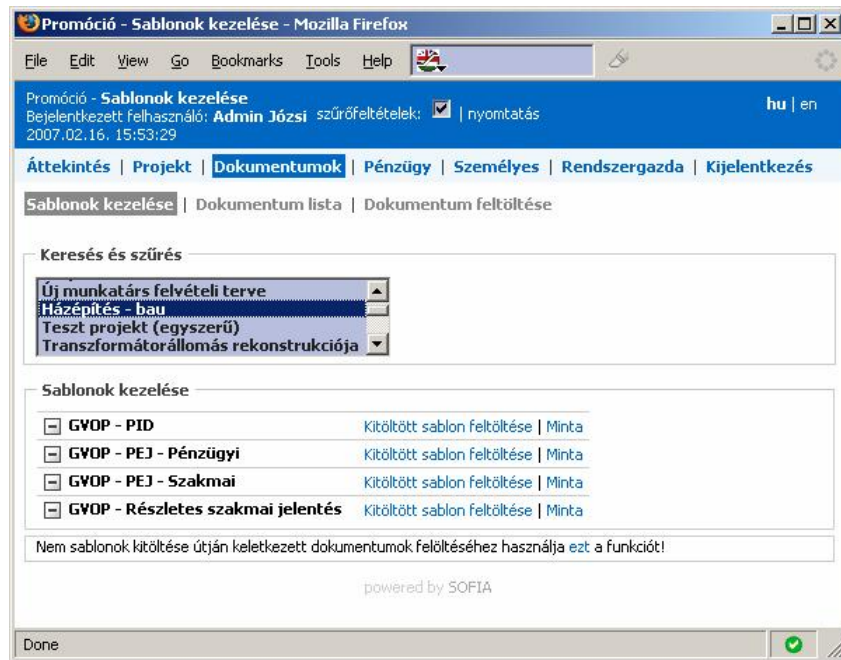
Ennek egyrészt az arvato systems Kft-nél hagyományos okai vannak, mivel ilyen megoldásokkal már a nyílt forrású változatok előtt is próbálkozott a cég. Másrészt a fent leírtakból látható, hogy ilyen mértékű integráltságot csak saját fejlesztésű rendszerrel lehet elérni. Például a közismert nyílt forráskódú Hibernate [6] perzisztencia kezelő rendszer nincs szervesen integrálva a Tapesrty [7] portálrendszerrel. Természetesen az összekapcsolás itt is megoldható, de mégsem annyira triviális, és a megoldás kevésbé flexibilis, mint egy saját fejlesztésű rendszer esetén.

3.3 Dokumentum menedzsment

A 2.1-es fejezetben kifejtésre kerültek a dokumentum kezelés azon buktatói, amivel portálfejlesztés illetve ügyviteli szoftverek készítése során szembesültünk. A “Promóció” projekt keretében ennek egy spektrumát tártuk fel, mégpedig a külső rendszerekben elkészült média anyagok tárolásának kérdését.

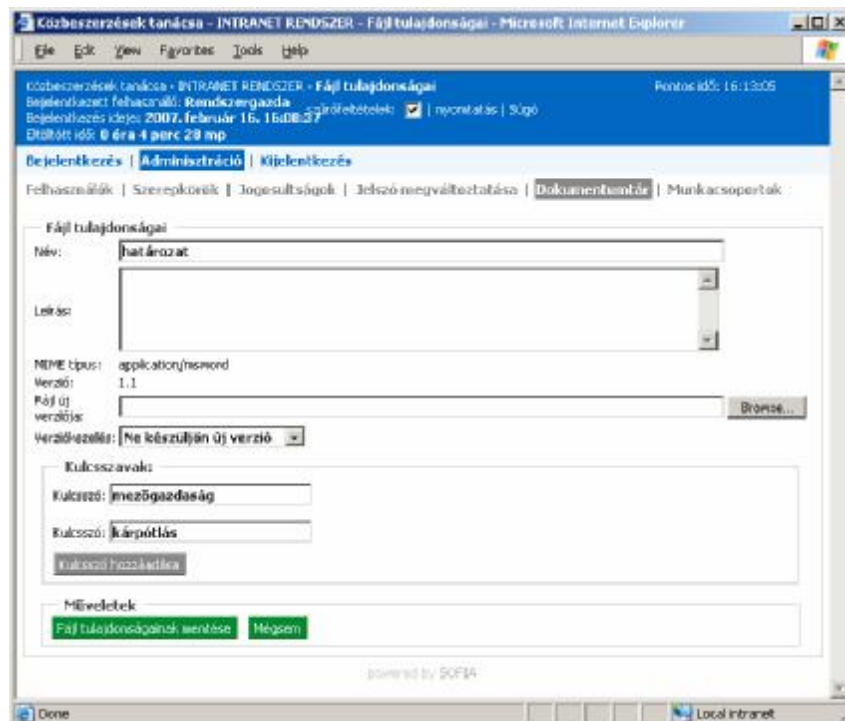
A Microsoft Word, vagy egyéb dokumentumokat a rendszer, mint állományként kezeli. Természetesen ebben az elgondolásban nem csak dokumentumok tárolhatók, hanem hanganyagok, képanyagok, továbbá a dokumentumokat formátumátum szerint is megkülönböztetjük, így például beszélhetünk Excel vagy PDF dokumentumokról. Ezeket összefoglaló néven Média anyagoknak nevezhetjük, és a formátumának megfelelően tartalom típusokba kategorizálhatjuk őket (MIME type).

A média állományok tipikusan feltöltés során kerülnek be a rendszerbe. Tehát, egy külső rendszerben (szoftverben) előáll az anyag, és ez kerül feltöltésre a dokumentum-menedzser rendszerbe. Jelen megvalósításban azonban a rendszer maga is képes dokumentumok előállítására, például a pályázat típusának megfelelő sablonok alapján – a pályázat lebonyolításának támogatása érdekében – a megfelelő projektlépésben a rendszer automatikusan dokumentumokat állít elő. A rendszerben lehet definiálni felhasználó által kitöltendő sablonokat is. A Promóció rendszerben a Projekt típusához lehet sablonokat rendelni, így a Projekt ügymenete által kívánt dokumentumok elkészítése a sablonok kitöltésével végezhető el. Lásd 1. ábra.



1. ábra: Sablonok alkalmazása

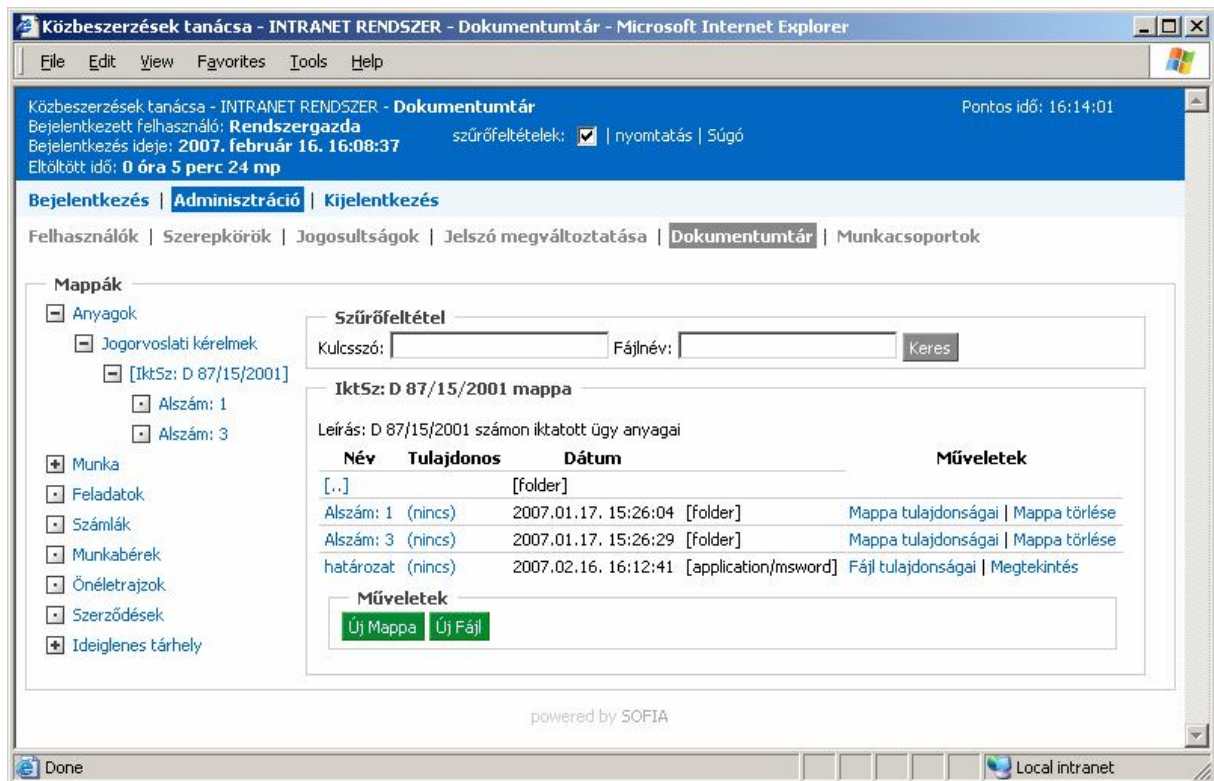
Az állományok a rendszerben felülírhatóak, azokból új verzió kerülhet feltöltésre. A feltöltött anyagok változáskövetését úgy oldottuk meg, hogy minden média anyag verziószámmal lett ellátva, és új verzió feltöltésekor ez a verziószám automatikusan növelhető. Lásd 2. ábra.



2. ábra: Feltöltött állomány részletei

A dokumentum menedzsment fontos ismérve az, hogy a dokumentumok strukturált, kereshető formában álljanak elő. A strukturáltság klasszikus formája a hierarchikus elrendezés

(gondoljunk például a fájlrendszerek könyvtárszerkezetére), viszont modernebb rendszerekben egyfajta kvázi-anarchikus megoldást támogatnak. Ez a megoldás a *címkézés*. Ebben az esetben az anyagokat egy címkehalmazból vett címkékkel tudom megjelölni, például témakör alapján. Ilyen formában a hasonló témakörű anyagok leválogatása a hasonló címkékkel rendelkező anyagok listázását jelenti. A címkék egyfajta ember által felruházott tulajdonságot jelölnek, a tulajdonsághalmazok definiálásának csak az emberi fantázia szab korlátot. Jelen megvalósításban egyfajta hibrid megoldást választottunk, itt nem lehet a címkéket előre definiált címkehalmazból választani, hanem azokat mindig esetileg kell megadni. Ezeket a címkéket mi keresőszavaknak neveztük, hiszen a dokumentumok utólagos keresése ezek alapján a metaadatok alapján lehetséges. Ezzel a megoldással egyúttal megadtuk a kivonatolás lehetőségét is, ugyanis egy dokumentum típusú média anyag kivonatolt szövegét beírhatjuk a keresőszavak mezőbe, az így megadott kivonatok alapján kereséskor természetesen megtaláljuk a témában érintett anyagokat. Mindezek mellett megtartottuk a klasszikus hierarchikus elrendezést, így a felhasználóknak nem fogadják idegenkedve az újszerű megoldást: Könyvtár-hierarchiát készíthetünk és az állományokat ezen könyvtárakba tölthetjük fel. Lásd 3.ábra.



3. ábra: Hierarchikus strukturálás és kereső felület

A megvalósított dokumentum kezelő modul szervesen használja a jogosultság kezelő modult, és egyes könyvtárak, vagy állományok hozzáférését korlátozhatjuk csoportokra, vagy felhasználókra.

4 Konklúzió

Egy olyan keretrendszer kifejlesztése, ami integrálja az adatbázis perzisztenciát és aktív támogatást nyújt a portál megvalósításhoz költségesebb egy nyílt forrású rendszer

alkalmazásához képest, de ez az egyszeri ráfordítás sokkal rugalmasabb, a fejlesztő cég profiljához jobban illeszkedő megoldást eredményez. Ez természetesen nem jelenti azt, hogy a nyílt forráskódú megoldásokat el kell vetni, hanem a meglévő megoldásokon alapuló innovatív ötleteket kell megvalósítani.

A projekt keretében fejlesztett portálrendszert a projekt lezárása óta fejleszti cégünk. Ma már AJAX technológiára alapozott komponens-modellt realizáló portálrendszer szolgál a fejlesztéseink alapjául.

A dokumentum kezelésben a nemrég szabványosított nyílt formátumok jelentenek majd áttörést, de ezek a megoldások sem fogják sosem pótolni a strukturált ügyviteli rendszerek funkcióját.

5 Köszönetnyilvánítás

Az itt említésre került portál fejlesztési ismeretek elmélyítésére nagy hatással volt az ORG projektnéven futó szintén GVOP támogatású Web alapú rendszer kifejlesztésekor szerzett tapasztalatok. Ez a rendszer a Magyar Onkológia Intézet felkérésére került elkészítésre, és jogszabályban előírt központi adatgyűjtést végez az egész ország területén megtalálható jelentésköteles intézményekből, mint elosztott rendszerből, illetve a beérkezett adatokra ellenőrzési és statisztikai lépéseket hajt végre.

A dokumentum menedzsmentben szerzett ismereteink spektrumát a KtPlusz rendszerünk készítésével bővítettük, ami már a bevezetésben említett web2-es AJAX technológiával kombinálva kerül kifejlesztésre.

Személyesen szeretnék köszönetet mondani Promóció projekt keretében Bertalan Tamásnak a projekt vezetéséért, Kroh Istvánnak a tervezésben és a sikeres lebonyolításban nyújtott szerepéért, és Garai Krisztiánnak a portál-keretrendszer szülőatyjának a kitartó munkájáért, és a fejlesztő csapat összes tagjának az áldozatos munkájáért.

6 Referenciák

1. BERTALAN Tamás, KOVÁCS László, MÁTÉTELKI Péter, MICSIK András, NÉMETH Géza, SCHILLINGNÉ HORVÁTH Ágota, TÓTH Zoltán: Visszacsatoláson alapuló intelligens projektmenedzsment, indikátorok felhasználásával. Gazdaságinformatikai Konferencia, GYŐR, 2006. november 10-11.
2. POI - <http://jakarta.apache.org/poi/>
3. Open Office XML - http://en.wikipedia.org/wiki/Office_Open_XML
4. Lucene - <http://lucene.apache.org/>
5. Velocity - <http://velocity.apache.org/>
6. Hibernate - <http://www.hibernate.org/>
7. Tapestry - <http://tapestry.apache.org/>