

# **Analýza možnosti využití CMS Drupal a repozitáře Fedora pro další rozvoj systému Kramerius, rozbor požadavků na administraci a seznam dalších možných rozšíření**

28. srpna 2009

M. Lhoták, M. Duda, R. Chýla  
Knihovna AV ČR, v.v.i.

Další vývoj systému Kramerius by měl svým účelem navazovat na stávající verzi 3.3.1, (k dispozici na <http://digit.lib.cas.cz/index.php?cat=downloads>). V rámci této analýzy se počítá s využitím open source repozitáře Fedora (<http://www.fedora-commons.org/> - release 3.2 a následující) jako jádra systému a CMS Drupal (<http://drupal.org/> - verze 6 a následující) jako uživatelského rozhraní. Nová verze Krameria, s rozlišovacím označením Kramerius 4, by měla vycházet z řešení nazvaného „Proof of Concept - využití repozitáře Fedora pro uložení digitálních objektů DTD Periodika a Monografie“ (<http://digit.lib.cas.cz/index.php?cat=fedora>).

Stávající verze systému Kramerius podporuje DTD Periodika a DTD Monografie (specifikace dostupná <http://kramerius-info.nkp.cz/odkazy%20a%20dokumenty/dokumenty/DTD/>). Nová verze bude podporovat formát metadat aplikovaný ve Fedoře při řešení výše uvedeného „Proof of Concept“, pro konverzi metadat ze stávajícího formátu bude využit konverzní nástroj vyvinutý také v rámci tohoto řešení. Je nutné dořešit podporu administrativních a technických metadat (<http://digit.nkp.cz/techstandards.html>).

Pro novou verzi je žádoucí přinejmenším zachovat stávající funkčnost systému Kramerius verze 3.3.1, přičemž musí být využit open source repozitář Fedora pro uložení dat. Zároveň je požadována podpora libovolných typů dokumentu, v první fázi zejména tzv. "nesloženého dokumentu" tedy dokumentu (i vícestránkového), který je obsahem jednoho souboru – článek, mapa, publikace, apod. (V dosavadní verzi DTD podporovaných systémem Kramerius je každá strana samostatným souborem a propojení do celku bylo zajištěno metadaty.) Tyto dokumenty systém umožní skládat do větších celků. Prostřednictvím OAI-PMH musí být dostupný profil, který je kompatibilní s projektem Europeana (zahrnující pouze volně dostupné dokumenty).

Přehled důležitých funkcí stávající verze systému:

- plnotextové vyhledávání
- vyhledávání v metadatech
- procházení v dokumentech stromovou strukturou
- vyhledávání podle datumu
- stránkování
- rozlišení plně dostupných dokumentů ve výsledcích
- zobrazení metadat a primárních dokumentů
- generování PDF dokumentů složených z více stran
- generování METS
- podpora perzistentního URL (URI)
- podpora grafického formátu DjVu
- statický export – vytvoření html verze na CD/DVD
- replikace – možnost přenosu dat mezi jednotlivými instalacemi
- česká a anglická verze
- nápověda

## **Požadavky na řešení – analýza využití Fedory a Drupalu :**

Vycházíme z předpokladu, že systém má využít sílu Drupalu a Fedory. To znamená, že některé věci, které Kramerius dnes umí, je možné (nutně) udělat jinak.

Při návrhu je nutno rozlišovat funkce Drupalu a funkce stroje Fedora – tj. rozdělení různých typů úloh, obecně Drupal má tvořit pouze „krásné“ rozhraní pro ovládání Fedory, nicméně ne všechny operace správy musí probíhat právě přes Drupal. Fedora umožňuje, aby ji ovládal prakticky jakýkoliv klient – tj. na dálku. Lze si dobře představit situaci, kdy vlastní import a správa dokumentů probíhá přes speciálního klienta, ale zobrazování dokumentů už obstarává Drupal (nicméně, výhodnější by bylo mít obě operace přístupné přes webové rozhraní, ne nutně v Drupalu, např. součástí nové verze Fedory už je webový klient, který má v budoucnu zcela nahradit Java klienta)

Pro účely tohoto textu tedy budeme rozlišovat front-end (Drupal) a back-end (Fedora)

Výchozí filosofie pro návrh front-end(u): Drupal jako takový je CMS, ale zároveň i framework pro vytváření webových aplikací. Drupal je vytvořen z modulů, které obohacují jeho funkcionalitu a tedy i Kramerius by měl následovat stejnou filosofii aplikace tvořené z modulů na principu Lega™. Jisté komponenty budou z hlediska systému Kramerius základní a klíčové (např. administrace digitálních objektů), ale jiné komponenty funkcionalitu pouze rozšiřují. Tedy, při skládání Krameria si každá jednotlivá instituce musí nainstalovat základní modul(y) a může **přidávat** libovolné další moduly (důležitý rozdíl spočívá je v tom, že administrátoři **mohou přidávat nové moduly, nejde o vypínání nepotřebných modulů.**)

### **Co nabízí Drupal automaticky (rekapitulace nejdůležitějších relevantních vlastností)**

- Šablonovací systém
  - Drupal logiku i prezentaci obsahu **důsledně odděluje** (současná verze Krameria má problém oddělit logiku a prezentaci dat - kód je namíchán s instrukcemi pro zobrazení)
    - Lze generovat výstup pro množství zařízení a formátů s použitím šablon (zobrazení pro web, RSS, XML pro harvestery, brouzdání pro mobilní zařízení apod.)
      - Z toho plyne, že vývojáři nové verze musí počítat se zobrazováním pro několik zařízení (to je bude nutit vymýšlet systém flexibilně, Drupal jim tu práci značně ulehčí).
    - Jednotlivé bloky obsahu lze libovolně přesouvat (zobrazení není definováno pouze v šablonách, ale administrátoři mohou bloky libovolně přeskupovat přes GUI)
    - Avšak je nutno pamatovat na následující:
      - Vlastní digitální dokumenty - objekty (datastreams) jsou uloženy ve Fedoře
      - Jejich formátování může obstarat Fedora (pomocí transformací), ale stejně tak mohou být transformovány vně Fedory, tj. na straně Drupalu
      - Tyto objekty musí být následně zakomponovány mezi ostatní obsah (tj. dovnitř šablony Drupalu)
      - Z toho plyne důležité rozdělení úloh – z hlediska uživatele se vše musí chovat jako jednotný a konzistentní systém, avšak z hlediska vývojářů jde vlastně o dvě různé operace pro účely zobrazení
        1. transformace objektů uložených ve Fedoře
        2. případné zakomponování objektů z bodu 1 do webu (Drupal)
  - Správa uživatelů
    - Drupal používá tzv role-based permissions, jednotliví uživatelé systému patří do určitých rolí (skupin oprávnění). Na základě toho, zda uživatel je, či není členem jisté skupiny, se pak následně rozlišuje, jaká oprávnění konkrétní uživatel má.
    - Přístupová práva mohou být velmi podrobná - systém rolí (skupin) a uživatelských účtů
      - Lze vytvářet libovolné role (v systému jsou 2 základní: anonymní a přihlášený uživatel) a přidávat jim oprávnění
      - Oprávnění lze kombinovat
        - Drupal automaticky vyřeší požadavek na možnost vytvořit v Krameriovi několik úrovní správců (a každé z této úrovní dát pouze určitá oprávnění)
      - Uživatelé nevidí, na co nemají právo (týká se i zobrazení určité stránky)
        - Např. administrátor uvidí na adrese /kramerius/index všechny možnosti, ale "pod-administrátor", i když se dostane na stejnou stránku, uvidí v ní pouze možnosti, ke kterým dostal oprávnění (např. pouze přidat dokument do indexu, ale nikoliv restartovat vyhledávací stroj)

- Systém obsahuje rozhraní pro správu uživatelských účtů (a množství modulů, které tuto správu ulehčují)
  - Vyhledávání a administrace uživatelů (včetně uživatelských profilů – informací o uživateli)
  - Zablokování uživatelů, mazání, zasílání zpráv apod.
  - Správa hesel, zapomenuté heslo, přihlašování přes proxy (např. EZProxy), systém centrálních identit (např. openID, Shibboleth)
- Správa oprávnění k digitálním objektům (object-based permissions)
  - Výše uvedený systém rolí nelze účinně použít pro správu digitálních objektů (pouze říká, na co má daný uživatel právo v rámci Drupalu)
  - Drupal obsahuje mechanismus pro nastavení oprávnění k jednotlivým objektům, s jeho pomocí by se dalo řešit problémy typu: „počítače z IP adresy mají přístup ke kolekci X, ale nikoliv ke kolekci Y“
    - Avšak Fedora má vlastní mechanismus (XACML), který je určen pro stejné účely
    - Lze doporučit, aby Drupal byl využit pro správu toho, co uživatelé mohou dělat uvnitř Drupalu, avšak aby Fedora byla použita ke kontrole oprávnění k samotným digitálním objektům. (např. uživatel zařazen do skupiny „administrators“ se dostane do administrační sekce Drupalu, avšak operace, které učiní, budou efektivní pouze nad objekty, ke kterým má právo – to má zajistit Fedora)
- Bezpečnost
  - Systém je velmi aktivně vyvíjen (bezpečnostní záplaty)
  - Automaticky upozorňuje na bezpečnostní updaty (a to i pro dodatečné moduly)
  - Drupal používá sessions a může fungovat pod http i https
  - Drupal je také velmi robustní (testují ho stovky vývojářů)
- API a skriptování
  - Manipulace obrázků apod.
    - Drupal obsahuje API pro manipulaci obrázků (jako pomocí knihoven GDI, tak externí aplikací ImageMagick).
    - Např. generování vodotisku nebo vytváření náhledů je velmi snadné díky API a dostupným knihovnám
  - V Drupalu je možné velmi snadno vytvářet skripty, které automaticky manipulují obsah (např. „ve všech ISSN nahraď pomlčku mezerou“)
    - Avšak systém také obsahuje moduly pro správu Drupalu přes XML-RPC, REST, SOAP
    - Pro repozitář to má vcelku zásadní vliv, protože
      - Pokud existuje jednotné API (a to existuje, viz Fedora)
      - Je možno mnoho jednorázových i opakujících se administrativních operací zautomatizovat
        - Ovládat jisté funkce Krameria přes příkazovou řádku
        - Napojit externí aplikace (v libovolném programovacím jazyce)
        - Připravit makra či zavést skriptovací jazyk do Drupalu (která provádí specifickou operaci a nic jiného)
        - A to bez nutnosti vyvíjet speciální moduly pro celkem jednoduché operace
  - Drupal obsahuje množství standardních modulů a aplikací třetích stran, které není potřeba pro potřeby Krameria vyvíjet (zde pouze výčet relevantních pro Krameria):
    - V jádru Drupalu
      - Lokalizace (tj. různé jazykové mutace rozhraní)
      - Podrobné logování a statistika
      - Přivěťivá URL (např. /journal-of-biology namísto /index.php?q=node/55)
      - Komentáře
      - RSS
      - XML-RPC
    - Vybrané zajímavé externí moduly:
      - Ajax a javascript knihovny (důležité pro Web 2.0)
      - CCK (vkládání dokumentů s libovolnou strukturou, flexibilní „content-types“)
      - Views (vytváření pohledů na data přes GUI rozhraní)
      - WYSIWYG editory
      - Captcha (obrana proti spamu a robotům)

## **Současné schopnosti Krameria**

Zde jsou pro srovnání uvedeny možnosti současného Krameria (webového rozhraní) a v závorce následující:

D – Drupal má řešení zakomponováno v sobě, není nutno nic měnit ani přidávat

F – Fedora má v sobě toto řešení

GSearch – externí modul pro stroj Fedora

D+F – existuje v obou systémech, ale bude potřeba je napojit

Pokud závorka chybí nebo obsahuje „?“, znamená to, že daná funkce současného Krameria není automaticky dostupná ať ve Fedoře či v Drupalu.

Editace hlavní stránky [D]

- Wysiwyg editor (ale pouze jedna stránka) [D]

Import dat (Data ingest) [F]

- Z xml souborů (různé módy – např. přepsat, přidat) [F]
- Základní historie (log – operací ingest) [D]

Hromadné úpravy (spuštění „skriptů“ nad neimportovanými daty)

- Např. "Replace \* by –", "Generate Persistent Identifiers"
- Preview operací

Replikace (?)

Statické importy (?)

Exporty [F+D]

Import slovníků (Import numerals) [D]

Správa fulltext indexů [F+D]

## **Uživatelské rozhraní**

Prohlížení

- Dle typu dokumentů (Periodical, Monographs) [F, GSearch]
- Obsahuje počty záznamů v jednotlivých částech [F, GSearch]
- Abecední rejstříky (názvů periodik), dále podle volumes, issue, pages [D]

Drobečková navigace nahoře v menu [D]

Vyhledávání [F+D]

- Základní a advanced [F+D]
- Výsledky vyhledávání rozlišeny (All, Periodicals, Monographs, Pages)

Navigace [D]

- Základní statické menu nahoře (odkazy Print, Help, Links, Česky) [D]
- Drobečková navigace pod ním [D]
- Levý sloupec – dynamicky (vlastně obsahuje jinak zobrazenou drobečkovou navigaci)

Prohlížení dokumentu (stránky)

- Levý sloupec
  - formulář pro posunování stránek (javascript dotaz)
  - funkce generování PDF
- Hlavní stránka (při zobrazení dokumentu)
  - Základní informace o straně (metadata)
  - Vespod djvu applet se zobrazenou stranou

Lokalizace [D]

- Jazyk rozhraní lze přepnout nahoře vpravo

## **Úkoly k řešení**

V této sekci jsou popsány základní problémy, které nový Kramerius musí řešit a nemůže se při tom spolehnout na existující řešení.

### **Vyhledávání:**

Drupal má v sobě modul pro integraci se SOLR, v případě Fedory je otázkou, zda využít SOLR či přímo nástroje dostupné ve Fedoře (Fedora má v sobě tzv. GenericSearch, který umí prohledávat jak

metadata, tak i fulltext jednotlivých objektů – v pozadí tohoto GSearch je několik indexovacích strojů, dostupných jako plugin, mezi nimi Lucene i Solr)

Doporučujeme, aby Kramerius využil GSearch, protože je to:

- standardní řešení
- podporované komunitou kolem Fedory
- je odladěno (např. změny v úložišti se automaticky promítají do indexu)
- podporuje SOLR

SOLR je velmi kvalitní nástroj pro vyhledávání (postavený na Lucene) a je určen pro napojení s externími aplikacemi (výsledky ale i např. indexování probíhá zasíláním xml). Výhodou takového řešení by bylo to, že indexovací stroj může indexovat i dokumenty, které nejsou součástí Fedory.

Otázkou je, jak vyřešit propojení mezi Drupalem a GSearch, tj.

- Drupal poskytuje formulář
- Zasílá dotaz SOLR(u)
- SOLR vrací XML výsledky
- Které Drupal zobrazuje uživateli
  - o Přitom (ideálně) dokáže výsledky vyhledávání:
    - Uložit
    - Vytisknout
    - Poslat dál
    - Kombinovat ve schránce
    - Exportovat apod.

Také je nutno najít snadný a jednoduchý mechanismus, jak aktualizovat index SOLR(u) obsahem, který není uložen ve Fedoře (obsah uložený ve Fedoře bude indexován automaticky).

Stav řešení:

- Drupal obsahuje více než 5 modulů, které umožňují napojit SOLR (drupal.org přitom používá apachesolr\_search)
- Islandora (projekt na propojení Drupalu a Fedory - <http://vre.uepi.ca/dev/islandora>) obsahuje vyhledávání které komunikuje s GSearch Fedory

## Prohlížení

Toto je problém související částečně s vyhledáváním. Spočívá v tom, že objekty jsou uloženy ve Fedoře, Drupal musí zobrazovat určité pohledy na tyto objekty (např. kolekce, subkolekce), dávat jakousi logickou a pochopitelnou strukturu – organizovat obsah. Nejde tedy pouze o prohlížení generovaných seznamů.

- Různé náhledy na seznam dokumentů
  - o Časopisy/rok vydání/ročník/číslo
  - o Klíčové slovo nebo téma
  - o Akademie/ústav/typ produkce
  - o Autor/články

Náhledy by přitom měly být dynamicky generované a to může představovat problém s rychlostí – pravděpodobně bude nutno hledat kompromis.

Stav řešení:

- Vyhledávací stroje mohou generovat určité rejstříky
- Drupal může cachovat určité rejstříky
- Nebo může každý dotaz sahat do Fedory

Lze doporučit, aby správa logické struktury kolekcí byla přenechána zcela stroji Fedora a Drupal byl využit pouze pro transformace výsledků do různých prezentačních podob.

## Požadavky na správu obsahu

V případě digitální knihovny, která primárně zpřístupňuje speciální typy dokumentů (např. vědecké články) je potřeba nezapomínat, že systém není omezen na tento „primární“ obsah, ale vedle něj se např. v různých částech publikují různé další informace administrativní povahy – např. krátké zprávy, nápověda k využívání systému, informace převzaté z jiných zdrojů apod. Přitom tento obsah je pouze připojen k primárnímu obsahu, digitální knihovna by bez něj mohla existovat, přitom však není potřeba ho archivovat či přidávat do kolekcí spravovaných strojem Fedora.

Je proto nutno rozlišovat mezi:

- digitálními objekty (uloženými v úložišti, tj. ve Fedoře)
- digitálními dokumenty (uloženými přímo v systému, např. novinky)

Publikování obecně:

- Systém musí mít flexibilní RSS feeds
- Typy dokumentů s libovolnou strukturou
  - Záleží na způsobu integrace, který zvolíme, nicméně je jisté, že v dig.knihovně budou prezentovány vedle sebe:
    - Současné typy dokumentů (viz DTD Krameria)
    - Další typy dokumentů (např. články, recenze, videa, obrázky – systém musí být připraven zpracovávat libovolně strukturovaný obsah)
    - Krátké zprávy
    - Informace o časopisech
  - Nápověda
  - Informace marketingové a administrativní povahy (např. „o nás“, odkazy na příslušné projekty)

Z hlediska správy obsahu je potřeba řešit (či využít existující řešení) co se týče

- transformací digitálních objektů uložených ve Fedoře pro Drupal
- XSLT šablony pro generování HTML/XML výstupů (z Fedory pro prezentaci na webu a jinde)
- webové formuláře pro vkládání dat (tj. obrácený tok, z Drupalu do Fedory, výstupem je validní FOXML)
- API pro správu dokumentů

Návrhy řešení okruhu těchto problémů budou mít zásadní význam při výběru dodavatele.

## Požadavky na podrobnou správu oprávnění k digitálním objektům

Je nutno rozlišovat práva:

- k správě systému (administrace funkcí systému)
- k digitálním objektům (zobrazení, prohlížení, úpravy objektů apod)

Drupal je schopen řešit zcela bod č. 1.

Drupal je schopen řešit i bod č. 2, ale existuje několik možných variant:

1. Drupal „access tables“ (to není externí modul, ale mechanismus přímo v jádře)
2. Externí GACL modul
3. XACML zabudovaný přímo do Fedory

Nicméně, je jisté, že musí existovat GUI pro snadnou správu práv k digitálním objektům. Proto je projekt třeba rozdělit do dvou sub-systémů:

1. Grafické rozhraní
2. Stroj/mechanismus správy oprávnění

Důrazně doporučujeme využít XACML zabudovaného přímo ve stroji Fedora a pro grafické rozhraní správy práv zvážit využití projektu Fedora Security Layer:

[http://www.fedora-commons.org/confluence/display/DEV/Fedora Enhanced Security Layer FESL Requirements](http://www.fedora-commons.org/confluence/display/DEV/Fedora+Enhanced+Security+Layer+FESL+Requirements).

Požadovány jsou také časové zámky pro oprávnění (např. otevřít přístup pro určitou skupinu, ale pravidlo automaticky smazat po uplynutí určitého data).

## Administrace systému

Veškerá nastavení systému by měla být konfigurovatelná přes web rozhraní (to se netýká XSLT šablon apod).

Co se týče správy digitálních objektů, je potřeba počítat s tím, že Fedora má svoje API pro správu objektů, takže množina operací, kterou by administrátor mohl chtít provozovat je menší. Nicméně i tak by bylo velmi vhodné mít terminálový či XML-RPC přístup k různým akcím knihovny (ovládat knihovnu na dálku – např. zadat příkaz aktualizuj metadata, hned poté: spusť indexování).

Následuje výčet vlastností systému, z nichž některé byly již zmíněny v předchozích částech:

### 1. Přístupová práva

V současné verzi systému je nedostačujícím způsobem řešena možnost přidělování přístupových práv k datovému repozitáři. Pokud chce správce systému zpřístupnit někomu neveřejný obsah, musí uvést do konfiguračního souboru jeho IP adresu, resp. IP adresu sítě, ze které uživatel přistupuje. Zde bohužel veškerá granularita přístupových práv končí a od okamžiku, kdy má uživatel povolen svůj adresní rozsah, může přistupovat nejen k původně požadovanému obsahu, ale v podstatě k celé neveřejné části repozitáře, což je evidentně nežádoucí. Z toho tedy plyne, že nová verze systému by měla mít výrazně propracovanější správu přístupových práv, která by umožnila podstatně jemnější nastavení.

#### 1.1. Uživatel

- možnost zakládat uživatelské účty s jednoznačnou identifikací
  - interní, např. náhodně generované číslo, kterým bude uživatel jednoznačně identifikovaný v systému, a které nebude moci ovlivnit ani správce
  - uživatelské jméno (jedinečné) a heslo
  - doména/seznam domén
  - IP adresa/seznam IP adres
  - identifikace fyzické osoby, tzn. adresa, e-mail, telefon, jméno apod.
  - aktivní/neaktivní (příznak pro zablokování účtu)
  - datum založení/modifikace účtu
  - možnost zapnout logování aktivity uživatelského účtu (statistiky)
  - možnost nastavení pořadí v jakém budou aplikovány jednotlivé stupně ověření s tím, že lze některé i vynechat/přeskočit
  - certifikát (to je už hodně těžký kalibr, doplňuji pro úplnost a zamyšlení)

Založit uživatelský účet by neměla být povinnost, ale např. jen volitelná možnost v případě, že by se přístupová práva neřešila prostřednictvím skupin - viz. dále popis skupin.

S takto uspořádaným uživatelským účtem má správce poměrně široké pole působnosti v tom, jak omezit přístupová práva.

Řekněme, že uživatel se bude do systému přihlašovat např. jen svým uživatelským jménem heslem. Systém by tedy měl někde obsahovat v uživatelském rozhraní možnost zalogování se. V okamžiku, kdy je uživatel úspěšně autentizovaný, bude si jeho internetový prohlížeč vyměňovat se systémem autentizační údaje - z toho tedy plyne nutnost implementovat podporu sessions. Uživatel bude v rámci této aktuální session jednoznačně identifikovaný svým uživatelským jménem (interně číslem) a bude již záležet na správci, jaká práva (myšleno práva k dokumentům) tomuto uživatelskému jménu přidělí. Zde by mohl správce skončit, nicméně se ještě rozhodne, že uživatele

dále omezí také IP adresou a doménou ze které přichází. Pro uživatele to bude znamenat, že se jednak bude muset přihlásit, ale také bude muset přicházet z důvěryhodné sítě identifikované jeho IP adresou a případně i doménou. Je také možné, že se uživatel nebude přihlašovat vůbec a bude ověřována jen jeho IP adresa, případně opět příslušnost k doméně.

## 1.2. Skupiny

- možnost zakládat skupiny v případě velkého počtu uživatelů s různými požadavky na přidělená práva k dokumentům v repozitáři
  - interní, např. náhodně generované číslo, kterým bude skupina jednoznačně identifikována v systému, a které nebude moci ovlivnit ani správce
  - jméno skupiny (jedinečné) a heslo
  - popis skupiny
  - seznam členů skupiny identifikovaných buď (vždy jen jedna varianta)
    - svým uživatelským účtem - jméno, resp. interní číslo (ostatní nastavení uživatelského účtu nebudou propagována do skupiny)
    - IP adresou/seznamem IP adres
    - doménou/seznamem domén
    - certifikátem (opět jen pro úvahu)

Založit skupinu by opět neměla být, jako v případě uživatelského účtu, povinnost, ale volitelná možnost.

V případě, že skupina bude obsahovat seznam uživatelských účtů, měla by opět existovat možnost zalogovat se v uživatelském rozhraní, ale tentokrát ne do uživatelského účtu ale do skupiny (otevřená zůstává otázka, zda-li zobrazovat seznam všech dostupných skupin aby si uživatel mohl vybrat - popis skupiny), prostřednictvím jména a hesla příslušné skupiny o které uživatel předpokládá/ví, že je jejím členem. Další záležitosti jsou již obdobné jako v případě jednotlivého uživatelského účtu.

## 1.3. Rozhraní pro správu a přidělování přístupových práv

Správce systému by měl mít možnost zakládat, modifikovat a rušit uživatelské účty a skupiny prostřednictvím administrátorského rozhraní v rozsahu popsaném výše bez nutnosti ručních modifikací konfiguračních souborů.

V okamžiku, kdy má správce vybudovanou strukturu uživatelských účtů, resp. skupin, musí mít také možnost propojit tuto strukturu s daty resp. metadaty v repozitáři. U jednotlivých objektů (obrázek, ročník, titul atd.), které jsou v rámci webového formuláře reprezentovány určitými prvky (odkazy, popisky, skrytá pole, session proměnné apod.) by měl existovat způsob výběru (vícenásobného) ze seznamu uživatelských účtů a skupin a provázání na příslušný objekt - zjednodušeně řečeno se jedná o vyplnění webového formuláře a odeslání do systému metodou POST. Interně, v rámci systému, by mělo být provázání mezi daty, resp. metadaty repozitáře a přístupovými právy realizováno prostřednictvím výše zmiňovaných interních čísel.

Dále by měla také existovat možnost (stejně jako ve staré verzi systému) nastavit dědičnost přístupových práv v rámci celého stromu objektů příslušného titulu. Pro autentizované uživatele se budou práva pravděpodobně dědit směrem dolů od přidělené úrovně bez ohledu na to, zda-li jsou dokumenty nastaveny jako neveřejné. Pro ostatní, anonymní uživatele, bude platit přístup stejný, který se používá ve stávající verzi systému.

## 2. Změny v "obecné" administraci

- Hlavní administrátor by měl mít možnost ovlivňovat podřízené administrátory v tom, co mohou v systému vidět a provádět. V současné verzi již takové možnosti jsou, ale bude potřeba je rozšířit, např. zda-li podřízený administrátor:
  - může generovat/mazat/aktualizovat fulltextový index
  - může generovat/mazat/aktualizovat statistiky
  - může modifikovat text na výchozí stránce
  - mazat ztracené soubory



- vidět importy a exporty ostatních administrátorů
- provádět hromadné úpravy
- apod. dle toho, jak se rozšíří funkcionality systému
- Administrátor by měl mít možnost
  - zastavit/opět spustit/zrušit právě probíhající import/export/replikací
  - smazat jemu příslušné záznamy v přehledech importů/exportů/replikací
  - třídit přehledy importů/exportů/replikací dle jednotlivých sloupců, např. dle data
  - vyhledávat v přehledech importů/exportů/replikací
  - maximum konfigurační práce provádět prostřednictvím formulářů, tedy webového rozhraní a nikoliv modifikací velkého množství konfiguračních souborů
  - snadné lokalizace ať již jazykové (moduly), ale hlavně také lokalizace pro cílovou instituci - tzn. šablony a binární kód by neměly obsahovat žádné konkrétní popisy, odkazy a pod. pro jednu určitou instituci
  - zapnout vyšší, upovídanejší, úroveň logování aplikace, což může výrazně přispět k identifikaci případných problémů

## Další zvažované změny a rozšíření funkčnosti

- zastavení importu a vyhledávání v exportech
- možnost editace metadat přímo v Krameriovi
- změnit přiřazování dokumentů (nyní staré k novým, po změně nové ke starým)
- zavedení vodotisku
- rozšíření tvorby .pdf a podpora více formátů
- zlepšit získávání DJVu plug-inu (pomocí Javy) + lépe zpracovat nápovědu
- u vícenásobných titulů možnost přímého listování mezi tituly jako je tomu stránek
- hromadné nastavení přístupnosti u titulů, ročníků a svazků
- různé struktury metadat – staré DTD i nové standardy
- generování náhledů
- zviditelnění technických a administrativních metadat
- zjednodušení listování
- propojení s image serverem
- možnost zobrazení každého el. dokumentu ve více kvalitách
- možnost uložení každého el. dokumentu ve více kvalitách
- rejstříky by měli podporovat i čísla a další znaky, které nejsou součástí české abecedy
- možnost různých voleb pro zobrazení, např. podle novinek, podle místa vydání, podle MDT apod.
- možnost voleb dalších písmen u abecedně řazených titulů, pokud se na stránce musí rolovat
- možnost listování po titulech bez nutnosti vracet se na vyšší úroveň kvůli výběru titulu
- pro automatizované zveřejňování podle datace (ochrana dle Autorského zákona) je potřeba kontrolovat další nakladatelské údaje o datu vydání, které jsou uvedeny v bibliografickém záznamu; nikoliv jak je tomu v současnosti, že se kontroluje jen pokud je jeden údaj a v případě každého dalšího údaje se záznam zveřejňuje ručně
- možnost zobrazení OCR textů
- rozšířit možnost exportů:
  - na všech úrovních nejen dokumentů, ale přednastavení širší volby např. djvu, txt, jpg..... a to hromadně (obrátit výběr) či jednotlivě
  - hromadné exporty titulů (opět s variantním rozšířením podle písmen, s jpg, txt, jen metadata; možnost volby úrovně)
  - nastavení preferované kvality - uvažujeme-li o různých kvalitách obrazových souborů, pak by uživatel měl mít možnost pohybovat se jen v takovém nastavení, které si vybere, aby nemusel neustále měnit požadavek
- možnost propojení se službou umožňující tisk knihy (publikace) na požádání (<http://www.librix.eu/cz/>)
- zobrazení informace o dostupnosti (volný dokument x dostupný pouze autorizovaným uživatelům nebo na určitém místě) už na úrovni listování podle autora nebo názvu dokumentu, nikoliv pouze u výsledků vyhledávání
- u periodik zobrazovat při listování podle názvu i informaci o časovém období, které je v Krameriovi přístupné (např. místo informace o počtu naskenovaných stránek)
- možnost přístupu nejen podle názvu periodika, ale i přímo podle konkrétního roku/měsíce/dne

- vyhledávání - možnost omezit vyhledávání pouze na konkrétní dokument a případně i datum (u periodik)
- výsledky vyhledávání - možnost dále třídit výsledky nejen podle typu dokumentu (monografie/periodikum), ale např. i podle názvu periodika nebo časového období
- nástroj pro zpětnou vazbu typu "hlášení chyb" např. u stránek, které jsou pomocí OCR chybně přečtené a další nástroje umožňující aktivně zapojit uživatele do zdokonalování obsahu digitální knihovny
- další případné rozšíření popisu digitálních dokumentů (např. možnost uvedení o jaký typ časopisu se jedná – „politický deník“, apod.)