

MASARYKOVA UNIVERZITA  
FAKULTA INFORMATIKY



# Šablonovací systém DSpace Manakin

BAKALÁŘSKÁ PRÁCE

**Tomáš Hofman**

Brno, jaro 2007

## **Prohlášení**

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

**Vedoucí práce:** Mgr. Vlastimil Krejčíř

## Shrnutí

Cílem této práce je prozkoumat a popsat možnosti šablonovacího systému DSpace Manakin, sloužícího k vytváření uživatelského rozhraní pro systém DSpace. Jako součást práce byly vytvořeny ukázky uživatelských rozhraní pro Digitální knihovnu fotografií Masarykovy univerzity [4] a pro projekt České digitální matematické knihovny [6].

## **Klíčová slova**

repozitář, DSpace, Manakin, xmlui2, xml, xslt

# Obsah

1	Úvod	1
2	<b>Systémy DSpace a DSpace Manakin</b>	<b>2</b>
2.1	<i>DSpace</i>	2
2.2	<i>DSpace Manakin</i>	2
3	<b>Stručný popis systému DSpace</b>	<b>4</b>
3.1	<i>Organizace dat</i>	4
3.2	<i>Popisná metadata v DSpace</i>	5
3.3	<i>Uživatelé v DSpace</i>	5
3.4	<i>Autorizace</i>	5
3.5	<i>Vkládání položek do systému DSpace</i>	6
3.6	<i>Aplikační rozhraní pro manipulaci s obsahem databáze DSpace</i>	6
3.7	<i>Závěr</i>	8
4	<b>Popis systému DSpace Manakin</b>	<b>9</b>
4.1	<i>Instalace systému Manakin</i>	9
4.2	<i>Architektura systému Manakin</i>	9
4.3	<i>Schéma DRI</i>	11
4.3.1	<i>Struktura schématu DRI</i>	11
4.3.1.1	<i>Element meta</i>	11
4.3.1.2	<i>Element options</i>	13
4.3.1.3	<i>Element body</i>	14
4.3.2	<i>Základní trojice atributů strukturních elementů</i>	14
4.3.3	<i>Slučování dokumentů DRI</i>	15
4.4	<i>Aspekty</i>	16
4.5	<i>Vzhledy</i>	19
4.6	<i>Lokalizace a internacionalizace</i>	19
5	<b>Příprava uživatelských rozhraní pro DKF MU a DML-CZ</b>	<b>21</b>
5.1	<i>Digitální knihovna fotografií Masarykovy univerzity (DKF MU)</i>	21
5.1.1	<i>Implementace</i>	21
5.1.1.1	<i>Úprava vzhledu</i>	22
5.1.1.2	<i>Navigační odkazy pro pohyb v hierarchii databáze systému DSpace</i>	22
5.1.1.3	<i>Režimy řazení a zobrazení stránek</i>	23
5.1.1.4	<i>Vkládání položek a administrace databáze systému DSpace</i>	24
5.2	<i>Česká digitální matematická knihovna (DML-CZ)</i>	24
5.2.1	<i>Implementace</i>	25
5.2.1.1	<i>Úprava vzhledu</i>	25
5.2.1.2	<i>Zobrazení referencí publikace</i>	25
5.2.1.3	<i>Zobrazení přehledu kódů MSC</i>	25
5.3	<i>Zhodnocení praktické části práce</i>	26
6	<b>Závěr</b>	<b>27</b>

Literatura . . . . .	29
A Ukázky vytvořených uživatelských rozhraní . . . . .	30

## Kapitola 1

### Úvod

Díky rostoucímu využití informačních technologií v moderním světě se stále větší množství dat ukládá v digitální podobě. Tento trend se bude v budoucnosti pravděpodobně prohlubovat, a tak vzniká potřeba vzrůstající množství digitálních dat vhodným způsobem organizovat a uchovávat. Za tímto účelem vzniklo v poslední době mnoho systémů určených pro archivaci a publikování nejrůznějších typů dat. Některé tyto systémy jsou úzce specializované na určitý typ dat, zatímco jiné jsou velice univerzální a lze je využít k archivaci prakticky jakýchkoli digitálních dat. Příkladem takového univerzálního systému je systém DSpace.

Robustnost systému DSpace je jeho velikou výhodou. Do jeho vývoje bylo investováno velké úsilí, díky čemuž může uspokojit velkou část požadavků, které dnes mohou být na archivační systém kladeny. Na druhou stranu však tato univerzálnost přináší velké množství problémů. Pracovní postupy jsou v systému DSpace často mnohem složitější, než by bylo v mnoha případech nasazení nutné, což zneprůjemňuje práci jeho uživatelům a systému to ubírá na přehlednosti a použitelnosti.

Tyto nedostatky by však často mohly být z velké části odstraněny, kdyby bylo možné systém DSpace snadno přizpůsobit podle požadavků konkrétní instituce. Tento problém se pokouší řešit systém Manakin, který je předmětem rozboru této práce. Systém Manakin poskytuje poměrně snadnou možnost, jak vytvořit uživatelské rozhraní schopné využívat všech možností systému DSpace, případně je ještě rozšířit o nové prvky, nebo naopak zakrýt nepotřebnou funkčnost. Systém Manakin umožňuje vybudovat prezentační vrstvu systému DSpace dle požadavků a potřeb uživatelů.

## Kapitola 2

### Systémy DSpace a DSpace Manakin

#### 2.1 DSpace

DSpace [10] [13] je systém pro ukládání, archivaci, indexování a prezentaci výzkumných dat, vyvíjený MIT Libraries a společností Hewlett-Packard. První verze systému DSpace byla uvolněna v roce 2002. Systém DSpace je vyvíjený pod licencí BSD. Využívá ho řada univerzit a výzkumných center po celém světě pro archivaci či publikování nejrůznějších dat. Na vývoji systému DSpace se mimo jiné podílí lidé z DSpace Federation, což je skupina institucí, které DSpace využívají.

Systém DSpace je napsán v jazyce Java a pro ukládání dat používá relační databáze - v současné době jsou podporovány PostgreSQL [25] a Oracle [24]. Pro prezentaci dat slouží primárně webové uživatelské rozhraní vybudované pomocí Java Server Pages [16]. Podporovány jsou i další protokoly jako OAI-PMH v2.0 [23].

#### 2.2 DSpace Manakin

DSpace Manakin [17] je systém umožňující vytvářet nové uživatelské rozhraní pro systém DSpace, vyvinutý na Texas A&M University. Ke své funkci využívá značkovací jazyk XML a příbuzné technologie. Jeho účelem je nahradit nedostatky původního uživatelského rozhraní systému DSpace. Hlavní výhodou systému Manakin je možnost snadno přizpůsobit vzhled a funkce systému DSpace podle potřeb jednotlivých organizací, jež ho využívají, a umožnit jeho snadnější integraci do existujících webových portálů. Systém Manakin neprovádí žádné zásahy do stávajícího uživatelského rozhraní systému DSpace, ale místo toho pracuje přímo s veřejným aplikačním rozhraním, které systém DSpace poskytuje. Obě uživatelská rozhraní tak mohou fungovat zároveň, nezávisle na sobě.

Mezi hlavní přednosti systému Manakin patří:

- možnost definovat každé komunitě a kolekci v systému DSpace vlastní vzhled,
- snadná internacionalizace a lokalizace,
- oddělení funkční logiky od grafického návrhu,
- poskytování alternativního uživatelského rozhraní bez nutnosti odstranit původní uživatelské rozhraní DSpace.



Předchůdcem systému Manakin bylo uživatelské rozhraní nazvané Moa, jehož vývoj byl s nástupem systému Manakin zastaven. Při návrhu systému Manakin se vycházelo ze zkušeností získaných vývojem systému Moa, díky čemuž se u systému Manakin podařilo dosáhnout větší modularity a lepší efektivity. Zatímco systém Moa využíval ke zpracování XML aplikační rozhraní DOM [8], Manakin je založen na aplikačním rozhraní SAX [26] a systému pro vývoj webových aplikací Apache Cocoon [2], který pomáhá udržet požadovanou modularitu.

## Kapitola 3

### Stručný popis systému DSpace

Systém DSpace je vyvíjený za účelem archivace a publikování prakticky libovolných dat. Tato univerzálnost je na jednu stranu jeho výhodou, na druhou stranu to znamená, že téměř nikomu DSpace nevyhovuje úplně a často by bylo třeba ho přizpůsobit na míru potřebám jednotlivých institucí. Tuto nevýhodu by měl řešit právě systém Manakin.

V této kapitole stručně rozebereme systém DSpace. Podrobněji se zaměříme na jeho objektový model a aplikační rozhraní součástí, se kterými pracuje systém Manakin.

#### 3.1 Organizace dat

Základní organizační jednotkou v systému DSpace je takzvaná komunita (*Community*). Komunity lze do sebe libovolně zanořovat. Původně byly zamýšleny jako prostředek k rozdělení systému DSpace podle jednotlivých oddělení nebo výzkumných center dané organizace, mohou však být využity pro hierarchickou reprezentaci i jiných struktur (například pro tematické rozdělení archivu fotografií).

Kromě vnořených komunit může každá komunita obsahovat kolekce (*Collection*), což jsou jednotky pro ukládání dat, která spolu nějakým způsobem souvisí. Každá kolekce může být navíc umístěna ve více komunitách.

Kolekce se skládají z jednotlivých položek (*Item*), což jsou základní archivační jednotky v systému DSpace. Položka může být v systému DSpace opět umístěna ve více kolekcích zároveň, ale každá položka má vždy jen jednu kolekci, která je jejím vlastníkem (*owning collection*).

Položky dále obsahují dva typy dat – metadata a takzvané svazky (*Bundle*), do kterých se vkládají datové soubory (*Bitstream*). Účelem svazků je seskupovat soubory, které spolu nějak úzce souvisí. Například soubor HTML a obrázky, na které se v souboru odkazuje, by měly být uloženy pohromadě v jednom svazku.

Ve výchozím nastavení systému DSpace mohou položky obsahovat některé z těchto pojmenovaných svazků:

- *ORIGINAL* – svazek s originálními datovými soubory,
- *THUMBNAILS* – svazek s vygenerovanými náhledy obrázků,
- *TEXT* – textová forma souborů ze svazku *ORIGINAL*, sloužící pro účely indexace,
- *LICENSE* – licence organizace, kterou autor položky při jejím vložení schválil.

### 3.2 Popisná metadata v DSpace

K popisu položek se v systému DSpace používá soubor metadatových prvků Dublin Core [3]. Ten umožňuje uchovávat metadata o autorovi, datech (vytvoření, publikování, vložení do databáze, ...), identifikátorech (ISBN, ISSN, URL, ISMN), popisy položek (abstrakt, popis, sponzorství, obsah), titul a jiné. Veškeré popisné informace lze navíc uchovávat v různých jazycích. Soubor metadatových prvků Dublin Core lze také rozšiřovat o vlastní popisné prvky. Systém DSpace také umožňuje definovat další soubory metadatových prvků.

Jednoduchá popisná data o kolekcích a komunitách jsou udržována přímo v databázi systému DSpace bez použití složitějších popisných schémat. Jedná se zejména o název a krátký popis kolekce či komunity nebo jejich logo.

### 3.3 Uživatelé v DSpace

Některé operace, jako je prohlížení databáze, mohou být v systému DSpace prováděny anonymně, bez nutnosti být přihlášen jako registrovaný uživatel. Aby však mohl uživatel přidávat nové položky nebo vykonávat administrační úkony, je nutné se autentizovat. Registrovaní uživatelé se v DSpace nazývají *e-people*. O každém uživateli (*e-person*) si systém DSpace udržuje informace jako jsou jméno, příjmení, email, heslo a zda se uživatel smí přihlásit přes webové rozhraní.

### 3.4 Autorizace

Pro účely autorizace se uživatelé v systému DSpace řadí do skupin (*Group*). Každý uživatel může být členem libovolného množství skupin. Existují dvě speciální skupiny pojmenované *administrators* a *anonymous*. Uživatelé, kteří jsou členy skupiny *administrators*, jsou oprávněni provádět v systému DSpace jakékoliv akce. Skupina *anonymous* zahrnuje všechny uživatele.

Skupině uživatelů můžeme přiřadit oprávnění k provádění určité akce na určitém objektu. V závislosti na typu objektu můžeme skupinám přiřazovat různé akce k těmto objektům. Dále uvádíme příklady možných akcí.

- *ADD / REMOVE* – Oprávnění k přidávání nebo odebírání vnořených objektů z rodičovských objektů (například odebírání podkomunit nebo kolekcí z nadřazených komunit, odebírání souborů ze svazků apod.).
- *READ* – Oprávnění k prohlížení objektu (lze přiřadit kolekcím, položkám nebo souborům).
- *WRITE* – Oprávnění k modifikaci objektu.

### 3.5 Vkládání položek do systému DSpace

Nové položky se mohou do systému DSpace vkládat dvěma standardními způsoby – hromadně pomocí speciální aplikace (*importer*) nebo pomocí webového rozhraní systému DSpace. V obou případech probíhá vkládání nové položky podle stejného schématu. Nejprve je vytvořen takzvaný *in progress submission* objekt. Podle toho, jaké je nastavení kolekce, ve které novou položku vytváříme, může být zahájen schvalovací proces (v terminologii DSpace nazývaný *workflow*). Během schvalovacího procesu je položka kontrolována správci dané kolekce, kteří ji buďto zamítnou a položka je vrácena uživateli, nebo její vložení schválí. Po úspěšném schválení přichází na řadu instalátor položek (*item installer*), který z *in progress submission* objektu vytvoří novou položku a tu vloží do databáze.

Schvalovací proces v systému DSpace může mít až tři fáze. Každé fázi je možné přiřadit skupinu uživatelů, kteří mají přijímané položky v této fázi kontrolovat a schvalovat. Vždy, když se nějaká přijímaná položka dostane do dané fáze schvalovacího procesu, jsou uživatelé skupiny, která je k této fázi přiřazena, informováni e-mailem o nové položce čekající na schválení. Pokud daná fáze nemá přiřazenu žádnou skupinu uživatelů, je tato fáze přeskočena a objekt přejde do fáze následující. Pokud kolekce nemá přiřazenu skupinu uživatelů k žádné ze tří schvalovacích fází, *in progress submission* objekty jsou ihned transformovány na nové položky a uloženy do databáze.

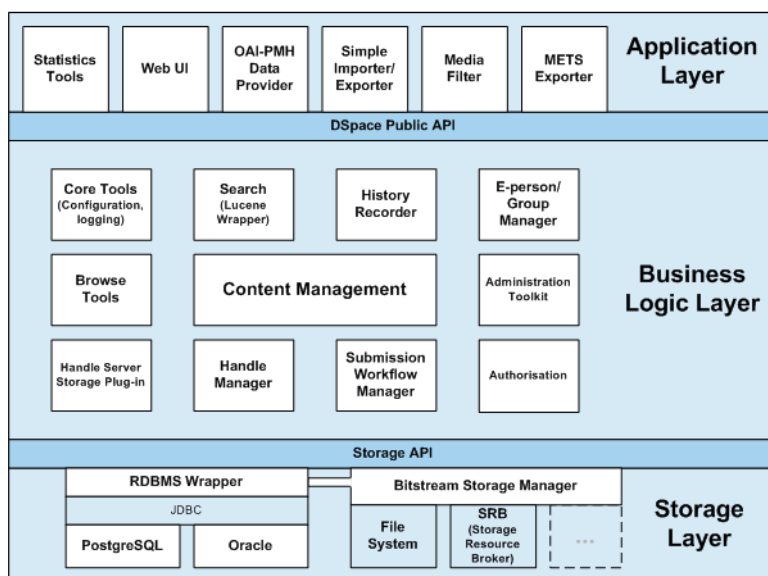
### 3.6 Aplikační rozhraní pro manipulaci s obsahem databáze DSpace

Architektura systému DSpace je rozdělena do tří vrstev – aplikační (*Application Layer*), logické (*Business Logic Layer*) a datové (*Storage Layer*). Logická vrstva komunikuje s datovou vrstvou pomocí rozhraní, které poskytuje datová vrstva, zatímco aplikační vrstva komunikuje s logickou vrstvou pomocí aplikačního rozhraní logické vrstvy. Protože systém Manakin nahrazuje aplikační vrstvu systému DSpace, je třeba se s aplikačním rozhraním logické vrstvy seznámit.

Všechny archivační objekty v DSpace (komunity, kolekce, položky, svazky a soubory) mají své reprezentace v objektovém modelu DSpace. Pro manipulaci a prohlížení komunit slouží třída *Community*, pro kolekce existuje třída *Collection*, a podobně existují třídy *Item*, *Bundle* a *Bitstream* pro položky, svazky, resp. soubory. Všechny zmíněné třídy se nacházejí v balíčku `org.dspace.content` a jsou potomky třídy *DSpaceObject* ze stejného balíčku.

Každá z těchto tříd obsahuje statické metody `find` sloužící k instanciaci objektů uložených v databázi. Konstruktory tříd jsou chráněné (`protected`), jednak proto, aby se zabránilo vytváření archivačních objektů samostatně, bez vazby k databázi systému DSpace, a také proto, aby nedocházelo k nesprávnému použití konstruktorů (vývojář by mohl předpokládat, že volání konstruktoru vytvoří nový objekt, zatímco ve skutečnosti by byl inicializován objekt existující, apod.). K vytváření nových objektů je tedy nutné používat metody typu `create` rodičovských objektů (například pro vytvoření nové kolekce zavoláme metodu `createCollection` rodičovské komunity) a pro instanciaci existujících objektů

### 3.6. APLIKAČNÍ ROZHRAŇÍ PRO MANIPULACI S OBSAHEM DATABÁZE DSPACE



Obrázek 3.1: Architektura systému DSpace

je nutné používat statické metody `find` příslušných tříd, nebo metody typu `get` rodičovských, případně dceřiných objektů (například pro získání položek kolekce použijeme metodu `getItems` této kolekce).

Použití metod `create` při vytváření nových objektů má tu výhodu, že si tyto metody samy zkontrolují, zda je přihlášený uživatel oprávněn vytvářet nové objekty v daném nadřazeném objektu.

Standardní postup při vytváření nové položky v systému DSpace se poněkud liší od vytváření ostatních objektů. Nejdříve je třeba vytvořit instanci třídy `WorkspaceItem`, které nastavíme všechna požadovaná metadata a vytvoříme v ní svazky (instance třídy `Bundle`), do kterých vložíme datové soubory. Jakmile máme instanci třídy `WorkspaceItem` připravenou, vytvoříme z ní pomocí statické metody `start` třídy `WorkflowManager` instanci třídy `Workflow`, čímž se zahájí schvalovací proces nové položky. Dále pak záleží na nastavení rodičovské kolekce, zda má k některé fázi schvalovacího procesu přiřazenu skupinu uživatelů. V případě, že ano, čeká instance třídy `Workflow` na schválení, po kterém postupuje do další fáze schvalovacího procesu. V opačném případě se z instance třídy `Workflow` rovnou vytvoří nová položka (instance třídy `Item`) a ta se uloží do databáze.

Při manipulaci s objekty v databázi systému DSpace je dobré mít na paměti, že některé změny se vykonávají jen v paměti a nejsou ihned provedeny v databázi. Mezi takové operace patří zejména změna metadat objektu. Aby byla změněná metadata skutečně uložena, je třeba volat metodu `update` příslušného objektu. Operace, které nějak souvisí se vztahy mezi objekty (například vytváření vnořených objektů v rodičovském objektu), jsou ihned prováděny v databázi.

Další třídou důležitou při manipulaci s objekty v databázi systému DSpace je třída `Con-`

text z balíčku `org.dspace.core`. Pokaždé, když chceme používat aplikační rozhraní pro přístup k objektům, je nutné nejdříve vytvořit instanci třídy `Context`. Ta je totiž vyžadována jako parametr ve většině metod pracujících s objekty systému DSpace. Při vytvoření instance třídy `Context` se mimo jiné otevře spojení s databází a zahájí nová databázová transakce. Veškeré změny, které váš kód v databázi provede, se uplatní až po zavolání metody `complete` příslušné instance třídy `Context`, která vyvolá ukončení transakce. Naopak, pokud by při běhu kódu došlo k nějakým chybám, všechny provedené změny mohou být zrušeny zavoláním metody `abort`, která současnou transakci zruší.

V instanci třídy `Context` je také uchována identifikace právě přihlášeného uživatele, aby mohla být prováděna autorizace prováděných operací.

### 3.7 Závěr

V této kapitole jsme stručně popsali systém DSpace a shrnuli jsme, jak jsou data v tomto systému organizována. Popsali jsme také způsob, jakým jsou data do systému DSpace vkládána a stručně jsme rozebrali některé nejdůležitější třídy pro manipulaci s daty v systému DSpace, jejichž znalost je nutná pro vytváření uživatelských rozhraní v systému Manakin a v této práci budou dále zmiňovány.

## Kapitola 4

### Popis systému DSpace Manakin

#### 4.1 Instalace systému Manakin

K nainstalování systému Manakin je třeba mít připraveny následující prerekvizity, na nichž funkce systému Manakin závisí:

- Java 5,
- Apache Ant 1.5,
- PostgreSQL,
- Apache Tomcat 5.5,
- DSpace 1.4.1 (a vyšší).

Samotný systém Manakin lze stáhnout buď jako *snapshot*<sup>1</sup> z oficiálních stránek systému Manakin, nebo si můžete obstarat nejaktuálnější verzi přímo z `svn` [29] repositáře projektu Manakin, což je doporučovaná cesta.

Po rozbalení zdrojových kódů je třeba zkopírovat konfigurační soubor systému Manakin `config/xmlui.conf` do adresáře s konfigurací systému DSpace. Následně Manakin zkompilujeme pomocí nástroje `ant` [1] a nainstalujeme ho na webový server Tomcat.

Podrobný popis instalace lze nalézt na [19].

#### 4.2 Architektura systému Manakin

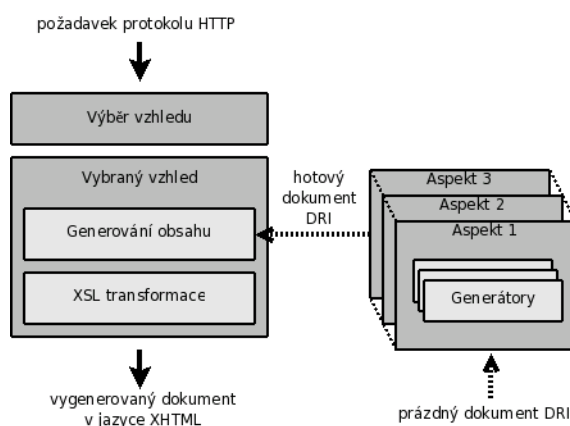
Systém Manakin se snaží v maximální možné míře využívat principu rozdělování úkolů (*Separation of Concerns* [28]). Tento přístup spočívá v rozdělení projektu do částí tak, aby se každá část zabývala jen svým přesně definovaným úkolem a nezasahovala do ostatních částí.

Architektura systému Manakin je rozdělena do tří vrstev, přičemž každá z vrstev je oddělena od ostatních a vyžaduje od vývojáře odlišné znalosti. To by mělo zvýšit efektivitu vývoje v systému Manakin, protože vývojář pracující v rámci jedné vrstvy nepotřebuje znalosti nutné pro vývoj v ostatních vrstvách.

---

1. Snapshot - Oficiální vydání počítačového systému. Obraz tohoto systému v určitém okamžiku jeho vývoje.

## 4.2. ARCHITEKTURA SYSTÉMU MANAKIN



Obrázek 4.1: Generování stránky v systému Manakin

### 1 Java / Cocoon vrstva

První vrstva se stará o vytvoření logického modelu požadované stránky. Tento logický model je reprezentován dokumentem XML ve formátu DRI (*Digital Repository Interface*), vyvinutého v rámci projektu Manakin. Na vytvoření reprezentace DRI každé stránky se podílejí takzvané aspekty, což jsou moduly, které se postupně volají a začleňují do vytvářeného dokumentu DRI vlastní obsah. Každý aspekt má svou specifickou funkci, např. prohlížení obsahu databáze systému DSpace nebo přidávání nových položek, a je naprosto nezávislý na ostatních aspektech. Při generování stránky dostanou možnost začlenit do stránky vlastní obsah vždy všechny aspekty přítomné v systému Manakin.

### 2 XML / XSL vrstva

V této vrstvě je na vytvořený DRI dokument aplikován vybraný vzhled (*theme*), který pomocí XSL transformací zajistí převod dokumentu DRI do požadovaného výstupního formátu. Typicky se tedy jedná o převod do formátu XHTML, ale do budoucna počítají vývojáři systému Manakin s vytvořením šablon pro převod do PDF nebo jiných formátů.

Zatímco všechny definované aspekty se aplikují na každou generovanou stránku, vzhled je vždy svázan s určitou komunitou nebo kolekcí. Při generování stránky se tedy vždy aplikuje jen jeden vzhled.

### 3 XHTML / CSS vrstva

V případě převodu do formátu XHTML přichází ke slovu ještě poslední vrstva, kde může vývojář použitím kaskádových stylů CSS ovlivnit zobrazení vygenerované stránky v prohlížeči.

Tomuto rozdělení do vrstev odpovídá také generování každé stránky v systému Manakin. Nejdříve je pomocí aspektů vygenerován dokument DRI popisující obsah stránky. Následně je zvolen vzhled, který se postará o transformování dokumentu DRI do výstupního



formátu – jazyka XHTML. Výsledný dokument v jazyce XHTML je potom zaslán uživateli. Celý postup je schematicky znázorněn na obrázku 4.1.

## 4.3 Schéma DRI

### 4.3.1 Struktura schématu DRI

Kořenový element dokumentu DRI `document` obsahuje tři dceřiné elementy – `body`, `options` a `meta`. Obsah těchto tří elementů si nyní stručně popíšeme.

#### 4.3.1.1 Element `meta`

Obsahuje všechna metadata o generované stránce, uživateli a obsahu části databáze, kterou si uživatel prohlíží. Tyto informace jsou uloženy ve třech dceřiných elementech `userMeta`, `pageMeta` a `objectMeta`. Narozdíl od zbylých dvou elementů neobsahuje element `meta` žádné informace o logické struktuře generované stránky.

Element `pageMeta` obsahuje informace o generované stránce, které jsou uloženy ve vnořených elementech `metadata`. Tento element obsahuje atribut `element`, popřípadě ještě atribut `qualifier`, říkající, co za informaci daný element `metadata` nese. Dále může obsahovat libovolné množství elementů `trail`, popisujících, kde se uživatel právě nachází v rámci systému. Typicky jsou v elementu `pageMeta` uloženy tyto informace:

- prohlížeč, který uživatel používá, jeho typ a verze,
- bazová adresa systému Manakin,
- nadpis stránky,
- informace o tom, zda má být uživatelův prohlížeč přesměrován na jinou stránku.

Element `userMeta` slouží k uchování informací o uživateli. Tyto informace jsou uloženy v dceřiných elementech `metadata`. Typicky se jedná o tyto informace:

- zda je uživatel přihlášený,
- identifikátor uživatele,
- jméno uživatele,
- adresa stránky pro přihlášení nebo odhlášení,
- adresa uživatelské stránky v systému,
- uživatelské preferované jazyky,
- přístupová práva uživatele.

Třetí skupinou metadat jsou informace o dostupných objektech databáze systému DSpace. Ty jsou umístěny v elementu `objectMeta`, kde jsou uloženy v dceřiných elementech `object`. Na elementy `object` může být následně odkazováno z elementu `body` (viz níže).

Pro názornost opět uvádíme jednoduchý příklad.

```
<document>
...

<meta>
  <pageMeta>
    <metadata element="contextPath">/manakin</metadata>
    <metadata element="title">DSpace Home</metadata>
    <trail target="/manakin">DSpace Home</trail>
    ...
  </pageMeta>

  <userMeta>
    <metadata element="identifier" qualifier="firstName">Franta</metadata>
    <metadata element="identifier" qualifier="lastName">Procházka</metadata>
    ...
  </userMeta>

  <objectMeta>
    <object objectIdentifier="hdl:123456789/81"
      url="/manakin/handle/123456789/81"
      repositoryIdentifier="hdl:123456789">
      <mets:METS LABEL="DSpace Community">
        ...
      </mets:METS>
    </object>
    ...
  </objectMeta>
</meta>

...
</document>
```

V uvedeném příkladě vidíme, že uvnitř elementu `pageMeta` jsou uloženy informace o báze adresy systému Manakin na daném webovém serveru (element `metadata` s atributem `element` obsahujícím hodnotu „contextPath“) a informace o nadpisu stránky (element `metadata` s hodnotou atributu `element` „title“). Jediný element `trail` nám říká, že se nacházíme na hlavní stránce systému Manakin. Pokud bychom se nacházeli uvnitř komunity na nejvyšší úrovni databáze systému DSpace, přibyl by zde další element `trail`

s názvem této komunity. Pokud bychom se nacházeli v komunitě vnořené uvnitř této komunity, přibyl by třetí element `trail` s názvem této vnořené komunity. Elementy `trail` tedy v podstatě popisují cestu k právě prohlíženému objektu.

V elementu `userMeta` jsou uloženy informace o právě přihlášeném uživateli, jehož jméno je Franta Procházka.

V posledním elementu pro ukládání metadat, `objectMeta`, vidíme informace o jednom objektu databáze systému DSpace – komunitě s identifikátorem „hdl:123456789/81“.

#### 4.3.1.2 Element `options`

Obsahuje všechny navigační odkazy a odkazy na akce, které je možné v rámci této stránky vykonat. Odkazy jsou uvnitř elementu `options` rozděleny do několika seznamů (elementy `list`), podle svého typu. Typy odkazů jsou následující:

- prohlížení,
- vyhledávání,
- výběr jazyka,
- neustále dostupné odkazy,
- odkazy závislé na kontextu (kontextem je zde myšlen právě prohlížený objekt, příkladem tohoto typu odkazu tedy může být odkaz na seznam všech položek v právě prohlížené komunitě apod.).

```
<document>
```

```
...
```

```
<options>
```

```
<list n="context">
```

```
<head>This Community</head>
```

```
<item>
```

```
<xref target="/manakin/handle/123456789/96/browse-title">
```

```
  By Titles</xref>
```

```
</item>
```

```
<item>
```

```
<xref target="/manakin/handle/123456789/96/browse-date">
```

```
  By Dates</xref>
```

```
</item>
```

```
...
```

```
</list>
```

```
</options>
```

```
...
</document>
```

V uvedeném příkladě vidíme právě seznam kontextových odkazů (element `list` s atributem `n` obsahujícím hodnotu „context“). Seznam obsahuje nadpis (element `head`) a dvě položky s odkazy – na stránku se seznamem položek této komunity seřazeným podle názvů položek a na stránku se seznamem položek seřazeným podle data vložení.

#### 4.3.1.3 Element `body`

Posledním top-level elementem je element `body`. Ten obsahuje veškeré informace o struktuře stránky, uložené ve strukturních elementech. To jsou například elementy `div`, dělící stránku na oddíly, které mohou dále obsahovat tabulky, odstavce, formuláře (ty jsou reprezentované jako elementy `div` s atributem `interactive`) a seznamy. Většina těchto elementů může mít také definován svůj nadpis pomocí elementu `head`. Uvnitř elementu `body` se také můžeme pomocí elementu `includeSet` odkazovat na metadata uložená v elementu `meta`.

```
<document>
...
<body>
  <div id="artifactbrowser.news" n="news" rend="primary">
    <head>DSpace Manakin is Live</head>
    <p rend="bold" n="welcome" id="artifactbrowser.welcome">
      Welcome to our new Manakin interface...
    </p>

    <includeSet rend="hierarchy" n="community-browser"
      id="artifactbrowser.community-browser">
      ...
    </includeSet>

    ...
  </div>
</body>

...
</document>
```

#### 4.3.2 Základní trojice atributů strukturních elementů

Většina strukturních elementů schématu DRI (`div`, `includeSet`, `list`, `p` a další) obsahuje standardní trojici atributů `id`, `rend` a `n` (viz elementy `div`, `p` a `includeSet` v předchozím

příkladě).

Atribut `rend` je vždy volitelný a může obsahovat několik mezerou oddělených hodnot, které mají sloužit jako pomůcka pro konečné formátování stránky. Zda a jak jsou atributy `rend` využity, záleží na aplikovaném vzhledu. Při použití standardních šablon pro převod formátu DRI do XHTML se hodnoty uložené v atributu `rend` většinou použijí v atributu `class` výsledných XHTML elementů.

Atribut `id` představuje jedinečný identifikátor v rámci celého dokumentu. Může být povinný nebo volitelný v závislosti na konkrétním elementu. Povinný je u všech elementů, které jsou nezávislé na svém kontextu, tedy například seznamy, či tabulky. Volitelný je naopak u elementů kontextově závislých, jako jsou jednotlivé buňky tabulky nebo položky seznamu. Hodnota tohoto atributu může být nastavena přímo aspektem, který daný element vytvořil, nebo je vygenerována automaticky podle Java Class Path<sup>2</sup> objektu, který element do stránky vložil.

Také poslední z této trojice, atribut `n`, slouží k identifikaci elementu. Tentokrát však neidentifikuje element v rámci celého dokumentu, ale jen v rámci jeho sourozenců. Reprezentuje tedy něco jako jméno elementu. Příkladem použití mohou být například položky seznamu, kde každá položka bude mít odlišné jméno (hodnotu atributu `n`), aby bylo možné je od sebe odlišit. V jiných seznamech v dokumentu však mohou být položky, jejichž jména jsou shodná s jmény položek v tomto seznamu.

### 4.3.3 Slučování dokumentů DRI

Již bylo řečeno, že na vytváření dokumentu DRI se v systému Manakin podílejí aspekty, které jsou jeden po druhém volány a začleňují do dokumentu vlastní prvky. Předtím než je aplikován první aspekt, připraví systém Manakin prázdný dokument DRI, který obsahuje jen kořenový element `document` a tři prázdné top-level elementy `body`, `options` a `meta`. První aplikovaný aspekt poté vytvoří vlastní dokument DRI, kam vloží vygenerovaný obsah a následně tento dokument sloučí s hlavním dokumentem DRI, který předtím připravil systém Manakin. Potom je aplikován další aspekt v pořadí, který opět vygeneruje svůj dokument DRI a sloučí ho s hlavním dokumentem. To se opakuje, dokud nejsou aplikovány všechny aspekty. Při slučování dokumentů DRI platí následující pravidla, popisující které elementy mohou být navzájem sloučeny a jak se naloží s jejich obsahem a dceřinými elementy.

Při slučování je vždy jeden dokument prohlášen za hlavní a druhý za doplňkový. Jejich elementy `body`, `options` a `meta` jsou potom zvlášť analyzovány a sloučeny.

Sloučení elementů `body` probíhá tak, že výsledný dokument bude ve svém elementu `body` obsahovat přesný obsah elementu `body` hlavního dokumentu, následovaný obsahem elementu `body` doplňkového dokumentu. Pokud se však v elementech `body` narazí na `div` se stejnými atributy `id` a `rend`, nebo na interaktivní `div` se stejnými atributy `action` a `method`, jsou tyto elementy sloučeny. Výsledný element `div` bude mít atributy `id`, `rend` a `n` stejné

2. Každý objekt v jazyce Java je instancí nějaké třídy, a každá třída je jednoznačně identifikována pomocí své Java Class Path. Ta se skládá z jména balíčku, do kterého třída patří, a jména samotné třídy.

jako `div` hlavního dokumentu a bude obsahovat všechny vnořené elementy z hlavního dokumentu následované vnořenými elementy doplňkového elementu. Pokud oba slučované elementy `div` opět obsahují `div` se stejnými atributy `id` a `rend`, budou tyto `div` elementy znovu sloučeny. Celý proces se takto rekurzivně opakuje, dokud se nepodaří sloučit všechny elementy `div`.

Při slučování elementů `options` se nejdříve porovnájí vnořené elementy `list` obou dokumentů. Ty které se nacházejí pouze v jednom z dokumentů, jsou jednoduše vloženy do výsledného dokumentu. Elementy `list` nacházející se v obou dokumentech (mají stejný `id` a `n` atribut) jsou sloučeny tak, že nový `list` obsahuje element `head` z hlavního dokumentu, následovaný položkami seznamu z hlavního dokumentu, za které se připojí položky seznamu z doplňkového dokumentu, které se liší od položek seznamu v hlavním dokumentu.

Slučování elementů `meta` je podobné, jako slučování elementů `body`. Dceřiné elementy `userMeta`, `pageMeta` a `objectMeta` jsou slučovány zvlášť tak, že obsah z doplňkového dokumentu je přidán za obsah hlavního dokumentu.

#### 4.4 Aspekty

Aspekty jsou implementovány jako sady transformátorů dokumentů DRI. Transformátory jsou objekty jazyka Java, které jsou potomky abstraktní třídy `AbstractDSpaceTransformer` z balíčku `org.dspace.app.xmlui`. Třída `AbstractDSpaceTransformer` má definované metody `addBody`, `addOptions`, `addPageMeta` a `addUserMeta`. Tyto metody lze ve vlastních transformátorech překrýt a vkládat pomocí nich do dokumentu DRI vlastní obsah. Ten je následně sloučen s hlavním dokumentem, podle pravidel popsaných výše.

Třída `AbstractDSpaceTransformer` také sama vytvoří novou instanci třídy `Context`, abychom mohli pracovat s databází systému `Dspace`.

Všechny elementy, které lze v dokumentu DRI vytvářet, jsou v jazyce Java reprezentovány třídami v balíčku `org.dspace.app.xmlui.wing.element`.

Nyní si uvedeme jednoduchý příklad transformátoru, který do generované stránky přidá odstavec se sdělením pro uživatele systému.

```
public class AddAnnouncement extends AbstractDSpaceTransformer {

    private static final String ANNOUNCEMENT = "Text sdělení...";

    public void addBody(Body body) {
        Division div = body.addDivision("announcement");
        div.setHead("Důležité sdělení");
        div.addPara(ANNOUNCEMENT);
    }
}
```

V příkladu je uvnitř elementu `body` vytvořen nový element `div` s atributem `n` nastaveným na hodnotu „announcement“. Dále je do tohoto elementu vložen nadpis „Důležité sdělení“

a nakonec ještě samotný text sdělení. Uvedený transformátor by vygeneroval následující dokument DRI.

```
<document>
  <meta/>
  <options/>

  <body>
    <div id="testingtransformer.div.announce" n="announcement">
      <head>Důležité sdělení</head>
      <p id="testingtransformer.para.announce">Text sdělení...</p>
    </div>
  </body>
</document>
```

Ve chvíli, kdy budeme mít vytvořeny vlastní transformátory, je třeba určit, kdy se mají jednotlivé transformátory aplikovat. To se provádí v konfiguračním souboru `sitemap.xml` v adresáři příslušného aspektu (`aspects/název_aspektu`). Struktura souborů `sitemap.xml` je definována schématem *Cocoon Sitemap*, detailně popsáným například na stránkách projektu Apache Cocoon [27]. Dostatečně dobře však lze soubory *Sitemap* pochopit i na základě studia těch z existujících aspektů systému Manakin.

Zde uvedu jen velice jednoduchý příklad, se kterým však jsme schopni vystačit ve většině běžných případů.

```
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    <map:transformers>
      <map:transformer name="Navigation" src="cz.muni.ics.xmlui.dkf.Navigation"/>
      <map:transformer name="BrowsingMode" src="cz.muni.ics.xmlui.dkf.Browsing"/>
      <map:transformer name="SubmitItem" src="cz.muni.ics.xmlui.dkf.Submission"/>
    </map:transformers>
  </map:components>

  <map:pipelines>
    <map:pipeline>

      <map:generate/>

      <map:match pattern="handle/*/*/dkfSubmitItem">
        <map:transform type="SubmitItem"/>
        <map:serialize type="xml"/>
      </map:match>
    </map:pipeline>
  </map:pipelines>
</map:sitemap>
```

```

    <map:transform type="Navigation"/>
    <map:transform type="BrowsingMode"/>

    <map:serialize type="xml"/>

  </map:pipeline>
</map:pipelines>
</map:sitemap>

```

Nejdříve je třeba definovat transformátory, které chceme v našem aspektu použít, což jsme provedli pomocí elementů `transformer`. Atributem `name` definujeme libovolný název, kterým se chceme na transformátor odkazovat, zatímco atribut `src` udává Java Class Path tohoto transformátoru.

Pravidla říkající kdy se mají jednotlivé transformátory aplikovat určíme uvnitř elementu `pipeline`. Element `pipeline` začíná vždy elementem `generate`, ten zajistí vytvoření prázdného dokumentu DRI, se kterým budeme dále pracovat. Elementy `transform` říkáme, že chceme na dokument DRI aplikovat příslušné generátory. Element `match` umožňuje větvit zpracování dokumentu DRI podle URL adresy požadované stránky, jejíž masku specifikujeme v atributu `pattern`. Transformátory nacházející se uvnitř elementu `match` (v našem případě transformátor `SubmitItem`) budou tedy aplikovány jen v případě, že adresa generované stránky odpovídá vzoru „handle/\*/\*/\*dkfSubmitItem“. V opačném případě se transformátor `SubmitItem` vynechá a aplikují se transformátory `Navigation` a `BrowsingMode`. Element `serialize` značí ukončení příslušné `pipeline`, jinými slovy ukončení zpracování dokumentu DRI. V případě, že by tedy adresa generované stránky v našem příkladě odpovídala vzoru v elementu `match`, aplikoval by se pouze transformátor `SubmitItem` a tím by zpracování dokumentu DRI v tomto aspektu skončilo.

Součástí systému Manakin je v současné době následujících pět hotových aspektů.

- *Artifact Browser* – slouží k prohlížení a vyhledávání v databázi systému DSpace.
- *E-person* – zajišťuje přihlašování a odhlašování uživatelů do systému, registraci nových uživatelů, editaci uživatelských profilů a změnu hesel.
- *Submission* – zajišťuje přidávání nových položek do DSpace, schvalovací proces a související úkony.
- *Administrative* – slouží k administračním úkonům, jako je vytváření a mazání komunit, kolekcí a uživatelů.
- *XML Test* – testovací aspekt demonstrující práci s dokumentem DRI.



## 4.5 Vzhledy

Vzhledy (*themes*) v systému Manakin se mohou skládat ze souborů šablon XSL a statických souborů, jako jsou obrázky nebo soubory kaskádových stylů. Šablony XSL mají za úkol transformovat vstupní dokument DRI do jazyka XHTML nebo jiného formátu, vhodného k odeslání uživateli. K tomuto účelu jsou v adresáři `themes/dri2xhtml/` připraveny již hotové transformační šablony pro převod z formátu DRI do jazyka XHTML, které lze importovat do nově vytvářeného vzhledu, takže se o převod do XHTML prakticky nemusíme starat. Tyto připravené transformační šablony jsou navíc poměrně dobře členěny. V případě, že by nám tedy výstup těchto šablon nevyhovoval, nemusíme vytvářet kompletní sadu vlastních šablon, ale stačí nám pouze předefinovat nevyhovující části těch připravených.

Uveďme si jednoduchý příklad. Řekněme, že chceme, aby stránky vygenerované naším vzhledem obsahovaly patičku s názvem univerzity. Ostatní části výsledného XHTML kódu chceme ponechat tak, jak je vygenerují připravené šablony. Pohledem do připravených šablon zjistíme, že o generování patičky stránek se stará šablona pojmenovaná *buildFooter*. Do souboru XSL našeho vzhledu tedy nejprve pomocí elementu `import` schématu XSL importujeme soubor s připravenými šablonami a následně předefinujeme šablonu *buildFooter*. Výsledný soubor XSL našeho vzhledu by potom mohl vypadat například takto:

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">

  <xsl:import href="../dri2xhtml.xsl"/>

  <xsl:template name="buildFooter">
    <div class="footer">
      (c) 2007 Masarykova univerzita
    </div>
  </xsl:template>

</xsl:stylesheet>
```

Podobně jako aspekty jsou vzhledy řízeny vlastními konfiguračními soubory *Cocoon Sitemap*. V těch lze specifikovat, kdy se má který transformační soubor použít nebo lze například provádět rozhodnutí, jaké soubory kaskádových stylů se mají do generované stránky načíst, v závislosti na typu prohlížeče používaného uživatelem. Opět se lze zorientovat pomocí poměrně dobře komentovaných konfiguračních souborů existujících vzhledů.

## 4.6 Lokalizace a internacionalizace

Snadná lokalizace a internacionalizace je jedním z hlavních požadavků při vývoji systému Manakin. Při vývoji uživatelského rozhraní v systému Manakin je klíčový zejména překlad použitých textů. Pokud nám na lokalizaci vyvíjeného uživatelského rozhraní záleží,

## 4.6. LOKALIZACE A INTERNACIONALIZACE

---

neměli bychom v průběhu generování žádné stránky našeho uživatelského rozhraní pracovat přímo s textem, který chceme do stránky vložit, ale místo toho použít *i18n* elementů, kterými systému Manakin řekneme, kam chceme vložit který text.

Pokud bychom tedy chtěli do stránky vložit například tlačítko „Přihlásit“, na místo popisku tlačítka vložíme element `i18n:text` jehož obsah bude klíč požadovaného textu v katalogu textů.

Výchozí katalog textů najdeme v souboru `i18n/messages.xml` v adresáři systému Manakin. Pokud následně chceme lokalizovat uživatelské rozhraní do nového jazyka, přidáme do adresáře `i18n/` nový katalog textů pojmenovaný `messages_lang.xml`, případně `messages_lang_country.xml`, s texty přeloženými do nového jazyka. *lang* a *country* v názvu souboru značí kód jazyka podle normy ISO 639-1 [15], resp. kód země podle normy ISO 3166 [14]. Pro český jazyk jsou to hodnoty „cs“ pro kód jazyka a „cz“ pro kód země. Není nutné překládat všechny texty z výchozího katalogu. Pro texty, které v lokalizovaném katalogu nebudou nalezeny, se použijí hodnoty z výchozího katalogu.

K nahrazení elementů *i18n* příslušnými texty dochází až v samotném závěru generování stránky, díky čemuž můžeme elementy *i18n* vkládat v kterékoliv fázi generování dokumentu – uvnitř aspektů vytvářejících dokument DRI, nebo v rámci vzhledu transformujícího dokument DRI do formátu XHTML.

## Kapitola 5

### Příprava uživatelských rozhraní pro DKF MU a DML-CZ

#### 5.1 Digitální knihovna fotografií Masarykovy univerzity (DKF MU)

DKF MU [4] je webový portál sloužící k archivaci fotografií. V současné době se jedná o aplikaci napsanou v jazyce PHP a vyvíjenou Ústavem výpočetní techniky Masarykovy univerzity.

Požadavky na uživatelské rozhraní DKF MU byly následující.

- Napodobit vzhled současné DKF MU.
- Pokusit se implementovat v systému Manakin některé prvky současné DKF MU, které v připravených vzhledech systému Manakin chybí. Jedná se zejména o navigační odkazy umožňující pohyb v hierarchii databáze systému DSpace (odkaz na následující objekt, předchozí objekt, rodičovský objekt apod.), přepínání řazení položek podle názvu, nebo podle data vložení a přepínání režimu zobrazení objektů v databázi systému DSpace mezi režimy „zobrazit jako seznam“ a „zobrazit v tabulce“.
- Vytvořit jednodušší systém pro vkládání nových položek a administraci databáze systému DSpace. Systém by měl být schopen vytvářet nové komunity a kolekce. Při vkládání nové fotografie by se měly automaticky vytvořit tři verze náhledů o různých rozměrech. Při prohlížení by měl mít uživatel možnost zvolit, jakou velikost fotografie chce zobrazit.

Dále byl ustanoven předpoklad, že pro DKF MU bude vyčleněna jedna komunita na nejvyšší úrovni systému DSpace Masarykovy univerzity, s čímž musí vytvářené uživatelské rozhraní počítat (alternativní možností by bylo dedikovat pro DKF MU samostatný systém DSpace). Vytvořené uživatelské rozhraní by tedy nemělo nijak zasahovat do ostatních komunit systému DSpace.

Ukázková instalace vytvořeného uživatelského rozhraní je k dispozici na [5].

##### 5.1.1 Implementace

Za účelem implementace nových funkcí byl vytvořen nový aspekt, nazvaný „DKFMU“, skládající se z několika nových transformátorů. Dále byl vytvořen nový vzhled, pojmenovaný „dkf“. Ten se stará o formátování výsledných stránek v jazyce XHTML.

## 5.1. DIGITÁLNÍ KNIHOVNA FOTOGRAFIÍ MASARYKOVY UNIVERZITY (DKF MU)

### 5.1.1.1 Úprava vzhledu

Klíčem k napodobení vzhledu současné DKF MU bylo předefinování několika šablon připravených transformačních souborů. Jednalo se zejména o šablony zajišťující základní rozvržení výstupní stránky XHTML a šablony pro převod metadat o zobrazovaných objektech. Bylo také třeba odříznout volání šablon zajišťujících převod navigačních odkazů ze sekce `options` v dokumentu DRI, protože tyto navigační odkazy v uživatelském rozhraní pro DKF MU zobrazovat nechceme.

### 5.1.1.2 Navigační odkazy pro pohyb v hierarchii databáze systému DSpace

Protože žádný ze standardních aspektů nevkládá do DRI dokumentu informace o sourozencích aktuálního objektu, bylo nutné vytvořit aspekt nový, který by tyto informace poskytoval. Bylo také nutné rozhodnout, jakým způsobem budou požadované informace do dokumentu DRI dodány. Logická volba by byla přidat adresy nových navigačních odkazů do sekce `options`. V tom případě by se ovšem odkazy začaly zobrazovat i ve všech ostatních komunitách, které při generování XHTML kódu používají připravené transformační šablony, protože tyto šablony do výsledné stránky vloží všechny odkazy nacházející se v elementu `options` dokumentu DRI. Protože aplikování aspektu nelze nijak jednoduše omezit pouze na vybrané komunity, musel by se tento problém řešit programově, uvnitř aspektu, kde by bylo nutné kontrolovat, zda se zobrazovaný objekt nachází v komunitě DKF a na základě toho rozhodnout, zda odkazy do stránky vložit či nikoli. Takové řešení by ale vyžadovalo dodatečnou konfiguraci, proto jsem raději zvolil druhou možnost – vkládat navigační odkazy do sekce `pageMeta`, kde je budou ostatní vzhledy ignorovat (element `pageMeta` schématu DRI totiž není připravenými šablonami převáděn do výsledné stránky celý, jako je tomu u elementu `options`, nýbrž jsou z něj použity jen určité informace).

Konkrétně byl vytvořen transformátor `Navigation` v balíčku `cz.muni.ics.xmlui.dkf`. V transformátoru byla vytvořena metoda `addCompassLinks`, volaná uvnitř metody `addPageMeta`, s následující funkcí:

Metoda nejprve ověří, jakého typu je právě procházený objekt. Následně získá rodičovský objekt a od něho zjistí seznam sourozenců právě zobrazovaného objektu. Seznam sourozenců je poté seřazen v závislosti na zvoleném řazení, což je další nově implementovaná funkce (viz dále). Nakonec jsou do `pageMeta` vloženy informace o adrese prvního, předchozího, následujícího a posledního sourozence a rodičovského objektu, pokud takové objekty existují.

Vytvořený transformátor do dokumentu DRI vloží následující fragment kódu.

```
<document>
...
<meta>
  <pageMeta>
    <metadata element="page" qualifier="parentObject">
      /manakin/handle/123456789/95
```

## 5.1. DIGITÁLNÍ KNIHOVNA FOTOGRAFIÍ MASARYKOVY UNIVERZITY (DKF MU)

---

```
</metadata>
<metadata element="page" qualifier="previousObject">
  /manakin/handle/123456789/97
</metadata>
<metadata element="page" qualifier="firstObject">
  /manakin/handle/123456789/98
</metadata>
</pageMeta>
</meta>
...
</document>
```

Dalším krokem bylo upravení transformačních šablon tak, aby byly tyto odkazy správně vloženy do generované stránky.

### 5.1.1.3 Režimy řazení a zobrazení stránek

Na stránce nového uživatelského rozhraní bylo vytvořeno několik grafických přepínačů, měnících způsob zobrazení stránky. Jedná se o tyto přepínače:

- Přepínač režimu prohlížení  
Přepíná mezi dvěma hodnotami – „normální“ a „kompaktní“. V režimu „kompaktní“ se na stránce nezobrazuje horní panel s nadpisem stránky.
- Přepínač režimu řazení  
Přepíná mezi dvěma hodnotami – řazení podle názvu objektů a řazení podle data vytvoření objektů.
- Přepínač způsobu zobrazení položek  
Přepíná mezi dvěma hodnotami – zobrazení v seznamu a zobrazení v tabulce.

Pro zajištění těchto funkcí byl vytvořen nový transformátor `BrowsingMode`. Transformátor je načítán do každé stránky a kontroluje, zda se v požadavku o stránku protokolu HTTP neobjeví parametry pro změnu některého z režimů. Aktuální stavy přepínačů ukládá do uživatelského kontejneru `SESSION`<sup>1</sup> a zároveň vkládá informace o stavech do elementu `pageMeta`. Tam si je při transformování dokumentu DRI přečte vytvořený vzhled a na jejich základě rozhodne, jakým způsobem stránku transformuje.

---

1. Webový server Tomcat si u každého návštěvníka udržuje kontejnerový objekt (jakési asociativní pole), do kterého lze ukládat informace týkající se tohoto návštěvníka. Tento kontejnerový objekt se často označuje termínem `SESSION`.

## 5.2. ČESKÁ DIGITÁLNÍ MATEMATICKÁ KNIHOVNA (DML-CZ)

---

### 5.1.1.4 Vkládání položek a administrace databáze systému DSpace

Toto byla časově asi nejnáročnější část uživatelského rozhraní pro DKF MU. Byly vytvořeny tři nové transformátory `SubmitItem`, `CreateCollection` a `CreateCommunity` se zhruba podobnou funkcí. Všechny vytváří samostatné stránky obsahující formulář s datovými poli nutnými pro vytvoření nového objektu. Transformátory vždy před vytvořením formuláře zkontrolují, zda již formulář nebyl odeslán a zda byla vyplněna všechna potřebná data. Pokud je toto splněno, vytvoří v databázi příslušný objekt a místo formuláře uživateli zobrazí odkaz zpět na stránku rodičovského objektu.

Poté bylo ještě nutné přidat do stránek odkazy na nově vytvořené formuláře. Tato funkce byla přidána to transformátoru `Navigation`. Zde se pomocí autorizačního systému implementovanému přímo v systému DSpace ověří, zda je uživatel oprávněn vytvářet v právě prohlíženém objektu nové dceřiné objekty a pokud ano, jsou do elementu `pageMeta` dokumentu DRI vloženy příslušné adresy.

## 5.2 Česká digitální matematická knihovna (DML-CZ)

DML-CZ [6] je projekt, jehož cílem je vyvinout a aplikovat nástroje umožňující vybudovat digitální knihovnu matematické literatury publikované v českých zemích. V současné době se díla převedená do digitalní podoby ukládají mimo jiné v systému DSpace. K jejich prezentaci se používá původní uživatelské rozhraní systému DSpace upravené přímým zásahem do kódu systému DSpace. To se však hůře udržuje a jeho rozvoj do budoucna by byl příliš náročný. Z těchto důvodů vznikl požadavek na vybudování alternativního uživatelského rozhraní pomocí systému Manakin.

Požadavky na nové uživatelské rozhraní byly následující:

- Vzhledově uživatelské rozhraní odlišit od standardního uživatelského rozhraní systému DSpace. Odstranit například některé nadpisy, které pro DML-CZ nemají význam.
- Vhodnou formou zajistit zobrazování metadat specifických pro DML-CZ, jako jsou kódy MSC<sup>2</sup> [22] vložených publikací, kódy publikací v některých matematických katalozích nebo seznam referencí vložených publikací.
- Vytvořit stránku, která umožní vhodným způsobem procházet kódy MSC všech vložených publikací, případně umožní snadno zobrazit všechny publikace patřící pod vybraný kód MSC.

Ukázková instalace vytvořeného uživatelského rozhraní je k dispozici na [7].

---

2. MSC – Mathematics Subject Classification

### 5.2.1 Implementace

#### 5.2.1.1 Úprava vzhledu

Při úpravě vzhledu bylo vytvořeno nové navigační menu, které nahradilo původní hlavní navigační menu systému Manakin. V novém navigačním menu přibyl odkaz na kompletní přehled kódů klasifikace MSC, jinak obsahuje téměř stejné položky jako menu původní, s tím rozdílem, že některé položky byly přejmenovány a u některých se změnila jejich cílová adresa na upravené verze některých stránek.

Mezi upravené stránky patří stránky zobrazující přehled názvů vložených publikací, přehled autorů a přehled kódů klasifikace MSC vložených publikací. U těchto stránek byly vytvořeny jejich nové verze, zobrazující větší počet položek, než původní stránky. U přehledu kódů MSC vložených publikací a přehledu autorů byl navíc upraven výstup těchto stránek tak, aby byly položky přehledu zobrazovány ve více sloupcích, což umožňuje efektivněji využít místo na stránce. Pro vytvoření upravených stránek byly využity upravené transformátory z původního aspektu *Artifact Browser*.

Dále byly upraveny některé šablony pro transformaci metadat, aby se u vložených publikací zobrazovaly jejich kódy v katalogu MathSciNet [21], případně v katalogu Zentralblattmath [30]. Kódy se navíc zobrazují ve formě odkazů na příslušné záznamy v těchto katalozích.

#### 5.2.1.2 Zobrazení referencí publikace

Součástí metadat o publikacích jsou také reference dané publikace. Ty jsou ke každé publikaci k dispozici ve formátu XML. Při importu publikace do systému DSpace se tento soubor XML přečte a informace o referencích se uloží v textové formě do souboru metadatových prvků Dublin Core dané položky. Reference by se tedy daly jednoduše zobrazovat z metadatových prvků Dublin Core. Problém je pouze v tom, že jelikož jsou již uloženy v textové formě, obtížně se formátují. Pro účely formátování by bylo mnohem vhodnější mít data o referencích v systému DSpace uložena i ve formátu XML.

Aby tedy bylo možné reference snadněji formátovat, rozhodli jsme, že spolu s daty publikace se bude do databáze systému DSpace ukládat i původní soubor XML obsahující informace o referencích. Pro tento soubor se vytvoří zvláštní svazek pojmenovaný *REFERENCES*. Ve vygenerovaných stránkách uživatelského rozhraní se zobrazují jen soubory ze svazku *ORIGINAL*, soubor s referencemi díky tomu uživatelé nevidí. Dále byl vytvořen nový transformátor, který soubory s referencemi ze systému DSpace načítá a vkládá do stránek zobrazujících publikace informace o referencích v naformátované podobě.

#### 5.2.1.3 Zobrazení přehledu kódů MSC

Pro klasifikaci článků v DML-CZ se používá klasifikační systém MSC2000 [22]. Jedná se o tříúrovňovou klasifikaci matematických oborů. Pro vytvoření stránky zobrazující přehled kódů MSC byl vytvořen nový transformátor, který do elementu `body` vytvářeného doku-

### 5.3. ZHODNOCENÍ PRAKTICKÉ ČÁSTI PRÁCE

---

mentu DRI vloží klasifikaci v podobě vnořených seznamů. Položky seznamů na nejnižší a prostřední úrovni klasifikace navíc nejsou tvořeny obyčejnými textovými uzly, ale odkazy, směřujícími na stránku, která vyhledá a zobrazí seznam publikací odpovídajících danému kódu MSC (ta již byla v systému Manakin implementována).

#### **5.3 Zhodnocení praktické části práce**

Účelem vyvinutých ukázek uživatelských rozhraní bylo především prozkoumat a demonstrovat možnosti systému Manakin. U projektu DML-CZ se nicméně počítá s vybudováním nového uživatelského rozhraní přizpůsobeného potřebám uživatelů projektu DML-CZ. Tato práce pomohla ukázat, že systém Manakin je pro tento úkol vhodný a je lepší alternativou, než přímá úprava kodů systému DSpace, což je způsob kterým bylo uživatelské rozhraní systému DSpace v projektu DML-CZ přizpůsobováno doposud.



## Kapitola 6

### Závěr

Cílem této práce bylo prozkoumat možnosti systému DSpace Manakin a pokusit se implementovat nové uživatelské rozhraní pro dva projekty Masarykovy univerzity – Digitální knihovnu fotografií Masarykovy univerzity a Českou digitální matematickou knihovnu. V obou případech se podařilo prokázat, že systém Manakin poskytuje dostatek prostředků ke splnění vytyčených cílů, nicméně prostor pro další úpravy zejména uživatelského rozhraní pro projekt České digitální matematické knihovny je poměrně široký a po konzultacích s uživateli tohoto projektu bude třeba ve vývoji tohoto uživatelského rozhraní pokračovat.

Systém Manakin umožnil zmíněná uživatelská rozhraní vybudovat relativně snadno, vezmeme-li v úvahu úsilí, které by bylo nutné vynaložit, pokud by se uživatelská rozhraní vyvíjela bez jeho pomoci. Vyvinutá uživatelská rozhraní jsou navíc mnohem lépe udržitelná, než pokud bychom upravovali původní uživatelské rozhraní systému DSpace. Velkou výhodou systému Manakin také je, že vytvořená uživatelská rozhraní jsou relativně snadno přenositelná na nové verze systému Manakin. Oproti tomu přenesení upraveného uživatelského rozhraní systému DSpace na novou verzi systému by mohlo být poměrně náročné.

Nevýhodou systému Manakin je jeho vysoká náročnost na výpočetní zdroje, ve srovnání s nároky samotného systému DSpace. To je především způsobeno použitím mladých technologií pro práci s formátem XML a lze očekávat, že v budoucnu zde dojde k výrazným optimalizacím. Systém Manakin je již nicméně využíván řadou institucí, což dokazuje, že na moderním počítačovém vybavení ho lze provozovat bez větších problémů.

Celkově lze systém Manakin hodnotit jako perspektivní projekt, který může výrazně zjednodušit přizpůsobování a rozšiřování uživatelského rozhraní systému DSpace, a jeho použití lze jedině doporučit.

## Literatura

- [1] Apache Ant, <<http://ant.apache.org/>> . 4.1
- [2] Apache Cocoon, <<http://cocoon.apache.org/>> . 2.2
- [3] Dublin Core, <<http://www.dublincore.org/>> . 3.2
- [4] DKF MU – Digitální knihovna fotografií Masarykovy univerzity, <<http://dkf.ics.muni.cz>> . (document), 5.1
- [5] Ukázková instalace vytvořeného uživatelského rozhraní k systému DKF MU, <<http://dl.ics.muni.cz:18080/manakin>> . 5.1
- [6] DML-CZ – Česká digitální matematická knihovna, <<http://dml.muni.cz>> . (document), 5.2
- [7] Ukázková instalace vytvořeného uživatelského rozhraní k systému DML-CZ, <<http://dl.ics.muni.cz:8080/manakin>> . 5.2
- [8] DOM – Document Object Model, <<http://www.w3.org/DOM/>> . 2.2
- [9] DRI Schema Reference, <<http://di.tamu.edu/projects/xmlui/schemaReference>> .
- [10] Dspace, <<http://www.dspace.org/>> . 2.1
- [11] DSpace System Documentation, <<http://dspace.org/technology/system-docs/>> .
- [12] DSpace Source Javadoc, <<http://www.mit.edu/people/mdiggory/javadoc/index.html>> .
- [13] České stránky o systému Dspace, <<http://www.dspace.org/>> . 2.1
- [14] ISO 3166, <<http://www.iso.org/iso/en/prods-services/iso3166ma/index.html>> . 4.6
- [15] ISO 639-1, <<http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=22109&ICS1=1&ICS2=140&ICS3=20>> . 4.6
- [16] Java Server Pages, <<http://java.sun.com/products/jsp/>> . 2.1
- [17] DSpace Manakin, <<http://di.tamu.edu/projects/xmlui/>> . 2.2
- [18] Manakin Developers Guide, <<http://di.tamu.edu/projects/xmlui/resources/DevelopersGuide.pdf>> .

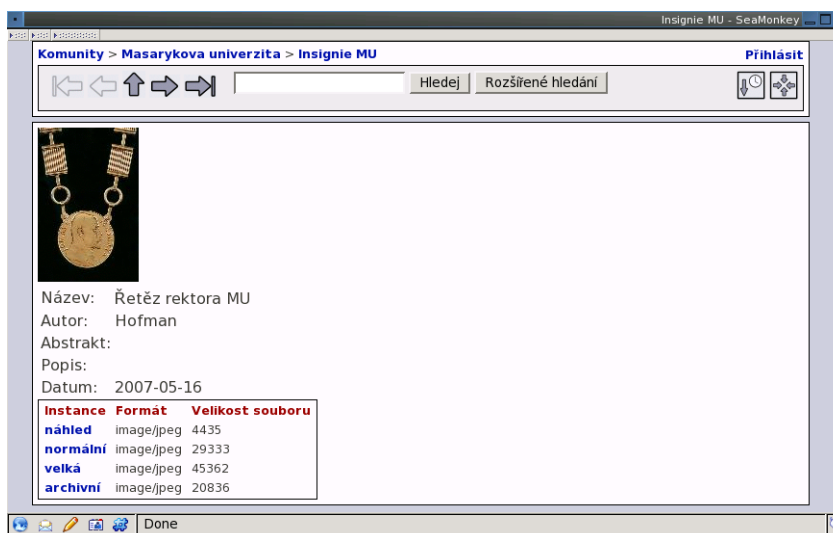
- 
- [19] *Manakin Install Documentation*, <<http://di.tamu.edu/projects/xmlui/install>>. 4.1
- [20] *Manakin Tutorial*, <<http://di.tamu.edu/projects/xmlui/resources/ThemeTutorial.pdf>>.
- [21] *MathSciNet – Mathematical Reviews on the Web*, <<http://www.ams.org/mathscinet/>>. 5.2.1.1
- [22] *MSC2000 – Mathematics Subject Classification 2000*, <<http://www.ams.org/msc/>>. 5.2, 5.2.1.3
- [23] *OAI-PMH v2.0 - Open Archives Initiative Protocol for Metadata Harvesting*, <<http://www.openarchives.org/OAI/openarchivesprotocol.html>>. 2.1
- [24] *Oracle*, <<http://www.oracle.com/database/>>. 2.1
- [25] *PostgreSQL*, <<http://www.postgresql.org/>>. 2.1
- [26] *SAX – Simple API for XML*, <<http://www.saxproject.org/>>. 2.2
- [27] *Cocoon Sitemap*, <<http://cocoon.apache.org/2.1/userdocs/concepts/sitemap.html>>. 4.4
- [28] *Separation of Concerns (SoC)*, <[http://trese.cs.utwente.nl/taosad/separation\\_of\\_concerns.htm](http://trese.cs.utwente.nl/taosad/separation_of_concerns.htm)>. 4.2
- [29] *Subversion*, <<http://subversion.tigris.org/>>. 4.1
- [30] *Zentralblatt Math*, <<http://www.zentralblatt-math.org/zmath/en/>> . 5.2.1.1

## Příloha A

### Ukázky vytvořených uživatelských rozhraní

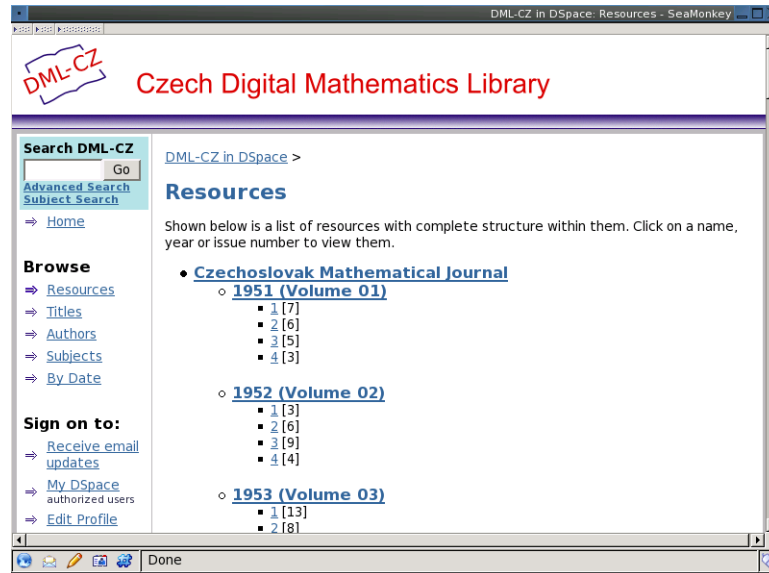


Obrázek A.1: DKF MU – zobrazení kolekce

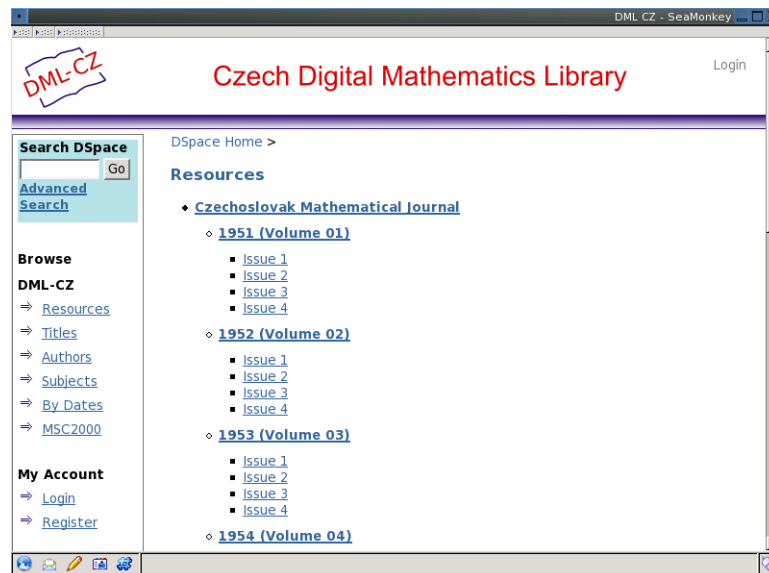


Obrázek A.2: DKF MU – zobrazení fotografie

## A. UKÁZKY VYTVOŘENÝCH UŽIVATELSKÝCH ROZHRAŇÍ

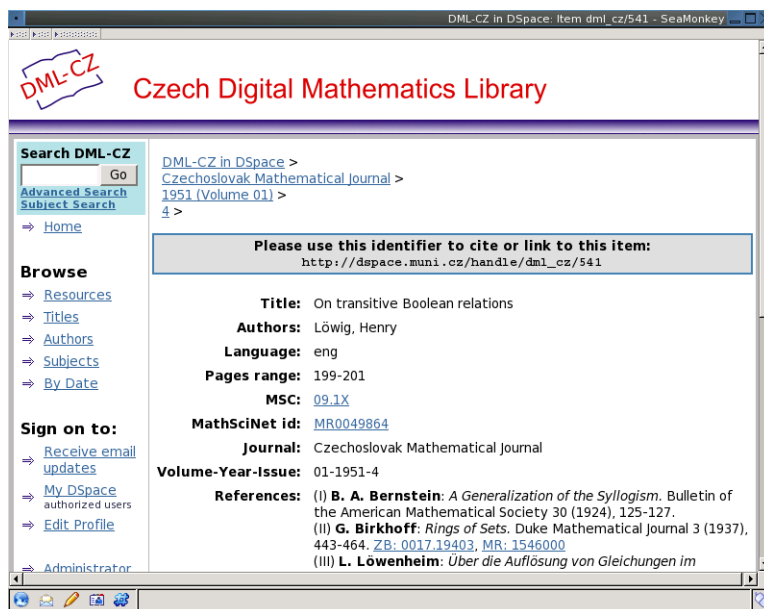


Obrázek A.3: DML-CZ – současné uživatelské rozhraní, přehled publikací

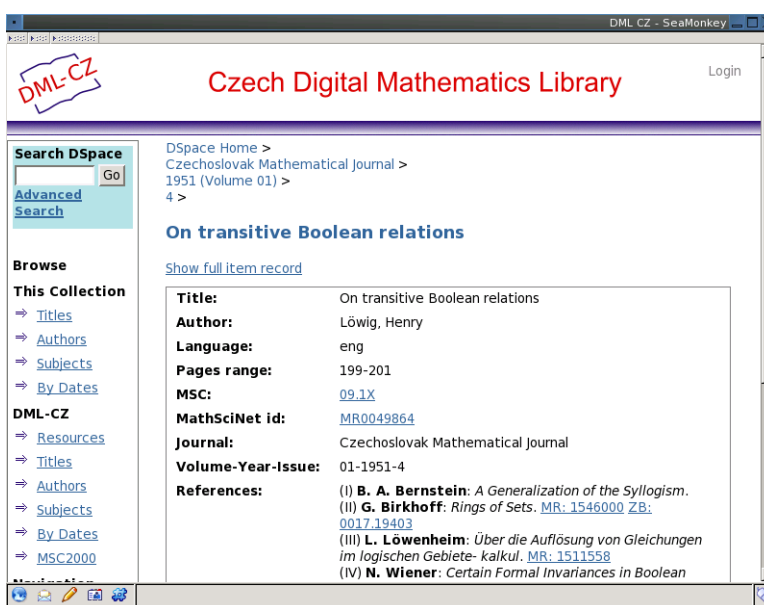


Obrázek A.4: DML-CZ – vytvořené uživatelské rozhraní, přehled publikací

## A. UKÁZKY VYTVOŘENÝCH UŽIVATELSKÝCH ROZHRAŇÍ

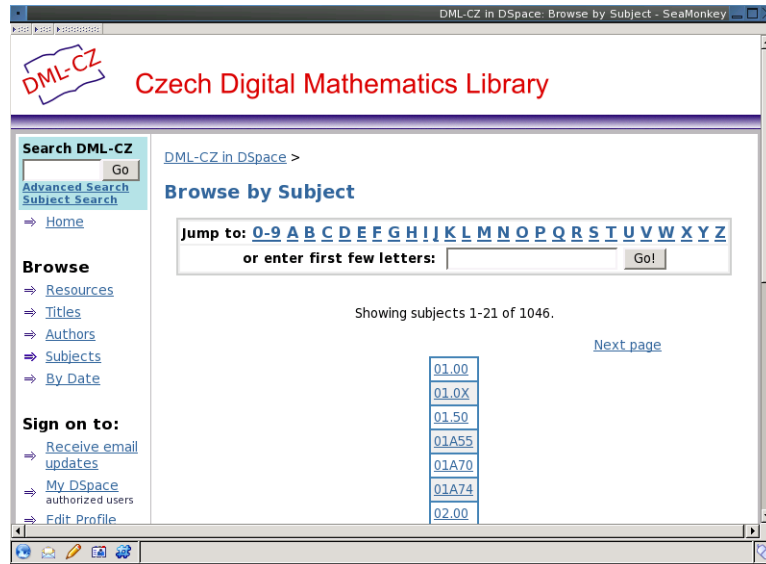


Obrázek A.5: DML-CZ – současné uživatelské rozhraní, zobrazení položky

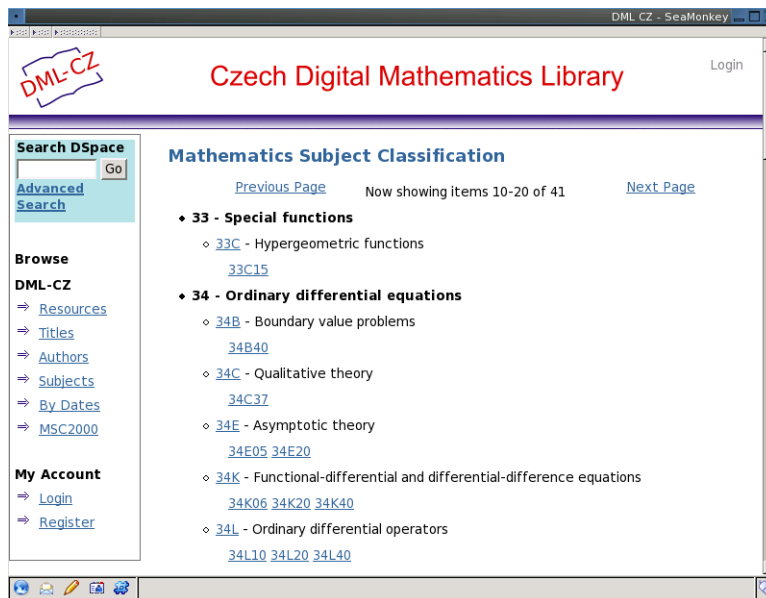


Obrázek A.6: DML-CZ – vytvořené uživatelské rozhraní, zobrazení položky

## A. UKÁZKY VYTVOŘENÝCH UŽIVATELSKÝCH ROZHRAŇÍ



Obrázek A.7: DML-CZ – současné uživatelské rozhraní, přehled kódů MSC



Obrázek A.8: DML-CZ – vytvořené uživatelské rozhraní, přehled kódů MSC