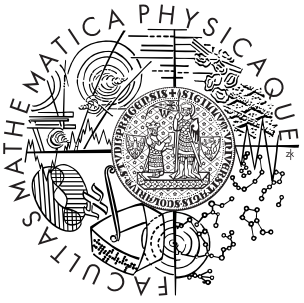
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Matyáš Novák

Zpřístupnění a správa naskenovaných dokumentů

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. Michal Žemlička, Ph.D.,
Studijní program: informatika, programování

2008

Tímto bych chtěl poděkovat vedoucímu své bakalářské práce, RNDr. Michalu Žemličkovi, Ph.D, za veškerý čas i pomoc, kterou mi věnoval.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 8. 8. 2008 Matyáš Novák

Obsah

Název práce: Zpřístupnění a správa naskenovaných dokumentů
Autor: Matyáš Novák
Katedra (ústav): Katedra softwarového inženýrství
Vedoucí bakalářské práce: RNDr. Michal Žemlička, Ph.D.
e-mail vedoucího: Michal.Zemlicka@mff.cuni.cz

Abstrakt: V této práci řešíme problém digitalizace převážně historických dokumentů. Cílem práce bylo navrhnout postup samotné digitalizace (scanování) a indexace těchto dokumentů se zaměřením na vnitřní strukturu dokumentů a na co nejsnažší vyhledávání v těchto dokumentech. Pro řešení problému bylo třeba navrhnout patřičné datové struktury a nástroje pro uložení digitalizovaných dokumentů i pro jejich katalogizaci, a provést analýzu, návrh a implementaci software potřebného pro provedení digitalizace i pro zveřejnění těchto dokumentů.

Navržené postupy a software byly ověřeny v pilotním projektu, ve kterém bylo dosud digitalizováno a (částečně) zkatalogizováno přibližně tři terabyty obrazových dat. Součástí práce je analýza zkušeností získaných tímto (stále běžícím) pilotním projektem. Dokumenty digitalizované během tohoto pilotního projektu jsou zveřejněny na internetové adrese `http://www.depositum.cz`.

Klíčová slova: digitalizace, knihovnictví, katalogizace, zpřístupnění historických dokumentů, digitální knihovna

Title: Accessing and Management of Scanned Documents
Author: Matyáš Novák
Department: Department of Software Engineering
Supervisor: RNDr. Michal Žemlička, Ph.D.
Supervisor's e-mail address: Michal.Zemlicka@mff.cuni.cz

Abstract: In this thesis we solve the problem of the digitization mainly historical documents. The goal of the thesis is to design a procedure for the digitization (scanning) and indexing these documents, with a focus on the internal structure of documents and easy searching in those documents. For a solution to the problem, it was necessary to propose appropriate structures and instruments for storing and cataloguing the digitized documents, and perform analysis, design and implementation of software required for the digitization process and for the publication of these documents.

The proposed procedures and software were validated in the pilot project. During this (still running) project about three terabytes of image data has been obtained and (partially) indexed. The thesis includes the analysis of the experiences gained from this pilot project. Documents that has been digitalized during this pilot project are available at `http://www.depositum.cz`.

Keywords: digitalization, librarianship, catalouging, digital library, cataloguing, disclosure of historical documents, digital library

Kapitola 1

Úvod

Cílem této práce je zpřístupnit historické dokumenty, převážně v podobě časopisů, pomocí interaktivni webové aplikace. Součástí práce je navrhnout pracovní postup digitalizace těchto dokumentů, navrhnout použité softwarové i hardwarové nástroje užité k digitalizaci; případně vyvinout potřebný software jak pro účely digitalizace, tak i samotného zveřejnění dokumentů.

Tato digitalizace probíhá na univerzitním pracovišti Katolické teologické fakulty Univerzity Karlovy. Zdigitalizované dokumenty jsou postupně uveřejňovány na internetové adrese `http://depositum.cz`.

V současné době probíhá digitalizace formou pilotního projektu. Vzhledem k měnícím se podmínkách zadání (získávání finančních prostředků či partnerů pro digitalizaci, postupný rozvoj projektu či naopak odkládání některých příliš finančně náročných kroků) není primárním cílem projektu vytvořit ucelené a hotové softwarové dílo, nýbrž navrhnout technologie a postupy pro provádění digitalizace a v nutné míře pro vyzkoušení užitých technologií vyvinout potřebné softwarové nástroje, vyzkoušet použité technologie a zhodnotit další možnosti rozvoje projektu.

Cílem projektu je tedy volba patřičných technologií, návrh pracovních postupů a vytvoření flexibilních a změnám otevřených softwarových nástrojů, který umožní co nejrychlejší pořízení prvních výsledků digitalizace, analýzu a vyhodnocení tohoto pilotního projektu a stanovení dalšího postupu a priorit při rozvoji projektu *Depozitum*.

1.1 Motivace

Knihovny (ať již univerzitní či jiné), archivy a další instituce mají k dispozici velké množství archivních materiálů, které mají značnou historickou cenu. Tato cena je ve dvou rovinách: cenná je samotná kniha či časopis jako historický předmět – často však ještě větší cenu mají informace v těchto dokumentech obsažené. Proto jsou tyto historické dokumenty většinou těžko přístupné (jsou k půjčení pouze prezenčně, či dokonce vůbec), neboť jejich zápůjčkou hrozí jednak zničení či poškození samotného dokumentu (časopisu, knihy…), jednak ztráta informací v ní obsažených.

I pokud je možné tento dokument zapůjčit, je zápůjčka zpravidla časově limitovaná, a její zapůjčení je pro vědeckého pracovníka spojeno se ztrátou času (cesta do knihovny a zpět). Tyto dokumenty jsou také většinou k dispozici pouze v jedné kopii, což například znemožňuje simultální práci dvou osob s jedním originálem. Ze starších dokumentů také často není povoleno pořizovat kopie, neboť nutná manipulace s dokumentem a obzvláště

prudký osvit při kopírování těmto dokumentům škodí.

1.1.1 Výhody digitalizace

Provede-li se digitalizace těchto dokumentů a zpřístupní-li se jejich digitální kopie na internetu, budou tyto dokumenty snadno k dispozici k výzkumu pro (prakticky) neomezenému počet lidí. Zároveň se velmi usnadní pořizování dalších kopií těchto dokumentů.

Digitalizací dokumentů se zároveň zajistí jejich ochrana a zachování pro budoucí generace, a to jak samotného historického předmětu (časopisu, knihy...), neboť s orginály nebude tak často manipulováno a budou tedy moci být bezpečně uloženy ve vyhovujícím prostředí, tak i informace, kterou dokumenty nesou: digitalizovaná data lze uchovávat ve velkém počtu kopií, což prakticky vylučuje jejich ztrátu.

Ke zdigitalizovaným dokumentům lze také pořídit rejstříky a opatřit je metadaty. Dále lze převést zdigitalizované dokumenty do textové podoby[1]. To vše umožní nejen rychlejší a pohodlnější přístup k informacím v těchto dokumentech, ale i možnost užít vědeckých metod, které by bez digitální podoby informací bylo velmi obtížné aplikovat (např. vytváření různých statistik, automatická textová analýza apod.), vyhledávat v nich pomocí fulltextového prohledávání a další možnosti.

[1] Textovou podobou myslíme textový formát – tzn. kódování informace po znacích např. ve formátu *unicode*, kdy jeden znak odpovídá jednomu písmenu.

Kapitola 2

Analýza a požadavky

V tomto oddílu práce popíšeme již existující projekty a jimi používané formáty. Pak se zaměříme na vlastnosti digitalizovaných dokumentů, jejich typické vlastnosti a specifika, a zároveň případné výjimky.

Zároveň popíšeme typického cílového uživatele systému *Depozitum*, jeho potřeby a tedy cíle, kterých chce tento projekt dosáhnout. V závěru kapitoly se pak budeme věnovat limitům, které vyplývají z omezeného rozpočtu, který je na tento projekt věnován a priority, které má tento projekt z těchto důvodů především naplňovat.

2.1 Existující knihovnické formáty a již existující projekty

V současné době existují tyto hlavní formáty pro výměnu knihovnických dat. Tyto formáty jsou určeny hlavně pro výměnu dat mezi jednotlivými knihovnami, popř. pro budování „nadknihovních katalogů“ zahrnujících zdroje z více knihoven. Není třeba, aby projekt *Depozitum* tyto standardy přímo implementovat, musí však být a je navržen tak, aby byl s těmito formáty (pokudmožno s přihlédnutím k jejich limitům) kompatibilní: tzn. aby informace pořizované pomocí systému *Depozitum* bylo v případě potřeby možno tyto formáty implementovat a komunikovat pomocí nich s dalšími subjekty provádějícími digitalizaci.

Dublin Core Spravované *Dublin Core Metadata Initiative* je jednoduchý, ale snadno rozšiřitelný formát pro popis zdrojů na webových stránkách, postupně rozšířený i pro indexaci dalších zdrojů [47, 20, 11]. Pro svoji přílišnou jednoduchost se však tento formát pro projekt *Depozitum* příliš nehodí, neboť s ním nelze zachytit dostatečné množství informací.

MARC Tento formát má dvě varianty serializace (zápisu informací uchovávaných pomocí tohoto formátu) – proto se spíše hovoří o dvou formátech. Tyto formáty však mají však stejné možnosti, liší se pouze formát, ve kterém informace uchovávají. Starší *MARC 21*[30, 10], užívá k serializaci vlastního formátu[1], novější formát *MARCXML* [31], který informace serializuje do *XML* [42] schématu.

[1] Tento formát používá kombinace textového a binárního zápisu dat.

Formát *MARC* je převážně zaměřen na popis fyzických dokumentů svazků (tzn. informací typu vydavatel, edice apod.), neposkytuje příliš možností pro popis obsahu svazku, ani možnosti pro propojování informací mezi jednotlivými záznamy.

TEI Formát *TEI* [29, 51] formát na bázi *XML* navržený pro účely doplnění textu o metadata (rozčlenění do kapitol, informace o zvýrazněných slovech, citacích apod.). Primárně tedy není určen k vytvoření separátních metadat popisujících digitalizované dokumenty. Poskytuje však možnosti pro uchování metadat popisujících dokument včetně jeho obsahu.

O užití tohoto formátu by bylo tedy vhodné uvažovat především při hledání vhodného formátu pro popis digitalizovaných dokumentů, převedených do textové podoby.

METS Formát *Metadata Encoding and Transmission Standard* [34, 32, 33] je určen k popisu dat obsažených v digitálních knihovnách (ať již digitalizovaných dokumentů, nebo archivovaných internetových stránek). Poskytuje možnosti pro zachycení struktury dokumentu, provázání struktury dokumentu s jednotlivými obrazovými soubory apod. Neposkytuje však možnosti pro detailnější popis jednotlivých entit této struktury (což je pro projekt *Depozitum* třeba). Do tohoto formátu je možno vložit i metadata v jiných formátech, např. přímo podporovaný je formát *MARC*.

Tento formát by měl být při vývoji projektu brán jako nutný základ – tzn. z projektu *Depozitum* by mělo být v případě potřeby možné generovat záznamy ve formátu *METS*.

MODS – Metadata Object Description Schema je *XML* schéma sloužící k popisu knihovních metadat. Je navržen jako rozšíření k *METS*, jako rozšíření *Dublin Core* a se zřetelem na kompatibilitu s formátem *MARC*[2]. Podobně jako formát *MARC* neposkytuje dostatečné možnosti pro popis obsahu dokumentů (popis jednotlivých částí dokumentů, vazeb mezi nimi apod.).

MASTER Formát *MASTER*[21, 3, 4] je formát (na bázi *XML*) pro popis obsahu dokumentů, včetně případných dalších metadat. Tento formát byl vyvíjen v Národní knihovně České republiky a je primárně cílen na středověké manuscripty – proto poskytuje v některých ohledech pro projekt depozitum zbytečné informace (informace o kvalitě rukopisu, použitých materiálech apod.). Tento formát byl dále rozšířen na formát *MASTER+*, který umožnuje uložená metadata provázat s obrazovými soubory digitalizovaných stran. Tímto formátem lze postihnout většinu dat, pořizovaných v rámci projektu *Depozitum*, přesto nelze některé skutečnosti zachytit, např. klíčová slova přiřazená k části dokumentu (článku), či návaznost textů mezi různými dokumenty apod.

Tento formát byl včeněn jako modul do verze 5 formátu *TEI* [28], vzhledem k tomu, že Národní knihovna ho však dále považuje za samostatný formát, a i jeho užití se poněkud liší od formátu TEI, do kterého byl začleněn, budeme ho považovat za samostatný formát.

[2]Není však s formátem *MARC* zcela kompatibilní, některé dle autorů nedůležité položky vypouští a některé strukturuje jiným způsobem. Dle autorů formátu je více orientovaný na uživatele systému a na vyhledávání.

Z39.50 Tento organizací *ANSI* standardizovaný protokol je určen k vyhledávání převážně) v internetových databázích [22] a je užíván některými nadknihovními katalogy pro souhrnné vyhledávání v katalozích knihoven [48]. V případě růstu projektu by bylo možno implementací tohoto protokolu zapojit projekt *Depozitum* do těchto nadknihovních katalogů.

V knihovním světě je dnes pravděpodobně nejužívanějším formátem formát *MARC*. Vzhledem k tomu, že s ním nelze postihnout všechna data, pořizovaná v rámci projektu *Depozitum*, jako nejvhodnější formát pro komunikaci s okolím se z těchto formátů jeví formát *MASTERS+*. Tento formát pokrývá největší část dat pořizovaných v rámci projektu *Depozitum* a je součástí světově uznávaného standardu *TEI*. Zároveň je používán Národní knihovnou České republiky, která je nejvýznamnějším českým subjektem provádějícím digitalizaci. Tento formát lze použít i pro přenášení textové podoby dat, popř je pro tato data možno použít standard *TEI*, který tvoří nadmnožinu tohoto formátu *MASTERS*.

2.1.1 Podobné projekty

Podobné zaměření, jako projekt *Depozitum*, mají v České republice následující (významnější) projekty digitalizace dat:

- *Národní digitální knihovna Národní knihovny České republiky*[3] provozuje dva systémy: *Manuscriptum*[4] pro staré rukopisy a spolu s Akademií věd České republiky systém *Kramérius*[5] pro periodika.
- Ústav pro českou literaturu České akademie věd má poměrně rozsáhlý archiv digitalizovaných literárních časopisů.[6]
- *Arcibiskupský zámek a zahrady v Kroměříži* digitalizují staré tisky ze svých knihovních fondů.[7]
- Zveřejnění digitalizovaných dokumentů chystá Městská knihovna v Praze (systém bude spuštěn v říjnu 2008).[8]
- Mnoho podobných systémů lze najít i v zahraničí.[9]

Tyto projekty jsou zaměřeny primárně na digitalizaci dokumentů. Přestože některé z nich poskytují mimo digitalizovaných dat i jejich anotace, hlavním cíl těchto projektů je zveřejnění daných knihovních jednotek. Až na výjimky se však nevěnují vnitřnímu členění rozsáhlých dokumentů, indexacím jejich obsahu aj. Obzvláště pro periodika, sborníky apod., kde se jeden svazek sestává z množství tématicky rozdílných částí, je však poskytovaná úroveň nedostatečná a uživatel je při vyhledávání konkrétních materiálů odkázán na sekvenční prohledávání možných zdrojů.

[3] `http://www.ndk.cz/narodni-dk`
[4] `http://www.manuscriptorium.com/Site/CZE/default_cze.asp`
[5] `http://kramerius.nkp.cz/kramerius/`
[6] `http://archiv.ucl.cas.cz/`
[7] `http://digi.azz.cz/`
[8] `http://http://www.mlp.cz/hispra.htm`
[9] např. `http://memory.loc.gov/`, `http://www.asztrik.hu/`, `http://www.library.uni.edu/instruction/digitalhistory.shtml`

Z českých projektů je tématicky nejblíže sytému *Depozitum* (alespoň co se týče pořizovaných dat) systém *Kramerius*. Ten však zobrazuje data pouze dle ročníků, čísel a stran, popřípadě poskytuje fulltextové vyhledávání pomocí neopraveného textu získaného pomocí OCR, neposkytuje však tématické vyhledávání dle klíčových slov, rejstříky ani rozsáhlejší anotace jednotlivých časopisových článků.

Projekt Národní knihovny *Manuscriptum* díky formátu *MASTER* (viz kapitola 2.1 na straně 9) se zaměřuje i na vnitřní strukturu dokumentů a je tak narozdíl od ostatních zde uvedených projektů schopen plnit všechny cíle, které si stanovuje projekt *Depozitum* (viz kapitola 3.1 na straně 21). Oproti projektu *Depositum* zaměřuje na dokumenty jiné povahy (většinou středověké manuscripty a další velmi cenné staré dokumenty), jednak zůstává primární filozofií projektu (či alespoň zveřejněného uživatelského rozhraní) zveřejnit samotné dokumenty, nikoli umožnit co nejsnáze vyhledat patřičné informace. Přestože je u dokumentů popsán jejich obsah, a vyhledávání umožňuje zobrazit dokumenty s daným obsahem, ovládací rozhraní neposkytuje možnost vyhledat přímo stranu či strany s daným textem.

V AiP Beroun s.r.o[10] probíhá pro Národní knihovnu vývoj nové generace tohoto projektu s názvem *Manuscriptorium verze 2.0* [1, 18, 2]. Tento projekt by měl uchovávat takzvaný *Komplexní digitální dokument*, což je soubor digitalizované dokumentu, textu získaného převodem dokumentu do textové podoby a metadat popisující tento dokument. Tato metadata budou pravděpodobně moci být ve formátech *MASTER+*, *TEI* a *METS*.

Archiv Ústavu pro Českou literaturu používá podobný systém jako Kramerius s o něco přívětivějším ovládacím a vyhledávacím rozhraním, ale bez možnosti fulltextového vyhledávání.

2.2 Členění, typy a vlastnosti digitalizovaných dokumentů

Projekt *Depozitum* je zaměřen na digitalizaci *dokumentů*.

Definice 1 Dokument *je libovolný logicky ucelený předmět či soubor předmětů určených k přenosu vizuální informace pomocí čtení či prohlížení.*

Dokumentem z pohledu projektu *Depozitum* tedy může být např. kniha, soubor hliněných destiček s navazujícím textem, či soubor jednotlivých čísel časopisů apod. Díky podmínce na logickou ucelenost nesplňuje definici dokumentu např. vytržených pět stran zprostřed knihy. Ze stejného důvodu v případě časopisu není za dokument považováno číslo či ročník, nýbrž soubor všech ročníků daného časopisu. V časopisech často vycházejí články na pokračování, polemiky aj., jednotlivá čísla či ročníky se tedy nedají považovat za zcela nezávislé a svébytné celky. Pohled na časopis jako na jeden komplexní dokument je tedy užitečný k podchycení struktury časopisu a vztahů mezi jednotlivými staťmi v časopise uveřejněnými (např. umožňuje snadno nalézt všechny recenze knih uveřejněné v libovolném čísle aj.).

Dokumenty, určené k digitalizaci, jsou většinou historické časopisy z devatenáctého a první poloviny dvacátého století. Prvním dokumentem, který je digitalizován, je *Časopis katolického duchovenstva*, ročníky 1828–1924 (dále *ČKD*), nyní jsou např. zahájeny

[10] `http://www.aipberoun.cz`

práce na *Českém časopisu historickém*, ročníky 1895–1989, časopise *Via 1968–1970*, *Život - exil* a dalších.Digitalizují se (či se o jejich digitalizaci uvažuje) však i dokumenty jiných typů, např. historický latinský slovník, soupis historických památek v Čechách z roku 1887, historické Bible aj.

V projektu *Depozitum* budou digitalizovány dokumenty různé povahy (časopisy, slovníky aj.), lze však předpokládat, že těchto typů dokumentů bude spíše malé množství a počet (různých) typů se bude zvětšovat velmi pomalu. Proto by nástroje systému *Depozitum* měly být specializovány na nyní digitalizované dokumenty (a tedy např. podchycovat i vlastnosti, které mají smysl pouze u tohoto typu dokumentu), mělo by však být zároveň snadno modifikovatelné pro jiný typ dokumentů, neboť v delším časovém horizontu se pak počítá s postupným rozrůzňováním typů digitalizovaných dokumentů.

Přestože budou digitalizovány různé druhy dokumentů, většina jejich rysů se shoduje, nebo je velmi podobná. Proto nyní budeme analyzovat „obecný dokument“, v dalších kapitolách se pak budeme věnovat specifikům jednotlivých dokumentů.

2.2.1 Členění dokumentů

Většina digitalizovaných dokumentů má vnitřní strukturu. Dokumenty lze členit dvěma způsoby: *fyzické členění* dokumentu vypovídá o struktuře jednotlivých fyzických částí dokumentu, zatímco *logické členění* podchycuje logické vazby mezi jednotlivými částmi textu. Tyto dva druhy členění odpovídají i standardu *METS* (viz kapitola 2.1 na straně 8).[11]

Fyzické členění – strany a svazky

Každý dokument lze členit na *strany*. Tento termín, stejně jako většinu termínů užitých v této práci je skoro nemožné definovat zároveň přesně a úplně, proto konečné rozhodnutí, co splňuje a nesplňuje danou definici, záleží na pracovníkovi řídícímu proces digitalizace. Primárním kritériem při aplikaci těchto definic není jejich doslovné dodržení, jako spíše zachování logické konzistence celého systému.

Definice 2 *Dokument lze disjunktně a úplně rozdělit na části, které se nazývají* strany. Strana *je část dokumentu, vytištěná (napsaný, nakreslený...) na jedné straně listu papíru, popř. jeho části definované ořezovými značkami.*

Lze-li takto definovanou část (disjunktně a úplně) rozdělit na souvislé (zpravidla obdélníkové) části, které jsou určeny pro tisk na samostatnou stranu, popř. je možno je tak vytisknout, aniž by byla porušena logika rozčlenění dokumentu do stran a nemají-li tyto části jinou souvislost než vztah návaznosti, je možno tyto části považovat za samostatné strany.

Strany budeme považovat za základní jednotku fyzického členění dokumentu, neboť jakékoli drobnější dělení nepodchycuje fyzickou strukturu dokumentu, pokud by tomu tak bylo, šlo by užít druhou větu z definice pojmu strana. Pokud je třeba zachytit jemnější strukturu než strany, je třeba užít *logického* členění, které budeme rozebírat v příští podkapitole.

Každý dokument má definované pořadí svých stran, definované pořadím stránek ve vazbě, datem vydání a logickou následností. Velká část dokumentů, má své strany fyzicky

[11] Element `structMap`: `http://www.loc.gov/standards/mets/docs/mets.v1-7.html#structMap`

rozdělené do skupin, které k sobě fyzicky náleží (svázáním, nebo následností) a zároveň tvoří logický celek (např. kniha se člení na jednotlivé díly, časopis na ročníky a ty na čísla...). Tyto části dokumentu budeme nazývat *svazky*.

Definice 3 *Souvislá posloupnost stran, která tvoří svoji strukturou jeden logický celek, budeme nazývat* svazek. *Každá strana dokumentu náleží alespoň do jednoho svazku, překrývají-li se dva svazky, pak je jeden svazek částí druhého.*

Podmínku souvislosti lze porušit, pokud je nesouvislost svazku zapříčiněna pouze vydavatelskými důvody a nenese podstatnou informaci.

Druhý odstavec definice je určen pro případ, kdy např. příloha časopisu vychází po částech v jednotlivých číslech jednoho ročníku. V tomto případě může být pro zachycení struktury dokumentu výhodnější tuto přílohu považovat za samostatný svazek zařazený na začátek či konec ročníku. Informace o fyzickém členění svazků je však (z hlediska knihovnictví) důležitý údaj. Proto by tato klauzule měla být užita pouze, pokud byla příloha zamýšlena k svázání dohromady, popř. pokud tento přístup k fyzickému členění svazku přinese větší výhody (např. ve snažší orientaci uživatele systému v digitalizovaném dokumentu), než uchování přesného pořadí stránek v originále.

Jelikož celý dokument je jeden svazek, a ten lze dělit na další svazky (např. časopis se dělí na ročníky a ty na čísla), tvoří svazky hierarchickou strukturu uspořádanou do stromu. Svazky na jedné úrovni stromu jdou dle seřadit dle návaznosti svazků, kompletní dokument lze získat průchodem stromu do hloubky (v každém vrcholu stromu se jeho děti procházejí v pořadí dle návaznosti).

Některé časopisy vycházeli po číslech, v knihovnách jsou však až na výjimky k dispozici ve formě svázaných ročníků. Proto není v požadavku na svazek požadována fyzická samostatnost svazku. Jelikož některé dokumenty jsou velmi rozsáhlé bez vnitřního faktického fyzického členění (např. kniha vydaná v jednom díle), přesto by ji bylo pro snažší orientaci uživatele užitečné považovat za fyzicky rozčleněnou. Proto lze za svazek považovat i oddělení či kapitolu dokumentu, byť nebyl nikdy fyzicky k dispozici jako samostatný dokument (vydán jako samostatný díl knihy apod.).

Přestože tato definice dává fyzickému členění určitou volnost (čímž umožňuje definovat fyzickou strukturu i u fyzicky monolitických dokumentů), musí fyzické členění dokumentu zachovávat organizaci, v jaké byl daný dokument vytištěn (s výjimkou zmíněných příloh). Nelze tedy např. v rámci fyzického členění prohazovat kapitoly apod. Zároveň by *svazek* měl nést ucelenou informaci. Proto by např. strana neměla být považována za svazek.

Logické členění – články

Druhou možností, jak pohlížet na strukturu dokumenty je *logické členění*. Logické členění sleduje logické vazby mezi jednotlivými částmi dokumentu, nezávisle na fyzické struktuře dokumentu. Logickým členěním lze tedy např. postihnout vztah textu a vysvětlivek uveřejněných na konci knihy, návaznost částí článku uveřejněného na pokračování ve více číslech časopisu apod.

Zatímco ve fyzickém členění tvoří základní jednotku strana, v logickém má tuto úlohu *základní článek*.

Definice 4 *Dokument lze disjunktně a úplně rozdělit na části, které se nazývají* základní články. Základní článek *je souvislá a logicky souvislá část dokumentu, nepřesahující hranice jednoho svazku.*

Pravidlo souvislosti lze porušit, pokud je kontinuita porušena jiným vloženým základním článkem malého rozsahu (např. sloupkem po straně časopisu, vloženou grafikou apod.).

Běžnými základními články jsou např. článek v časopise, postranní novinový sloupek, jedno heslo výkladového slovníku apod. Základní článek je v časopise běžně vymezen nadpisem (včetně) a nadpisem následujícího článku, případně koncem *svazku*. Za součást základního článku se považují i obrázky a jiné logicky k článku náležící texty či grafika (poznámky, schémata, dodatky apod.), pokud zachovávají jeho souvislost.

Pokud je to pro zachycení struktury vhodné, za základní článek lze považovat i část logicky souvislého textu, např. vysvětlivky uveřejněné na konci stati, kapitolu rozsáhlejšího článku apod. Za základními články se také považují titulní strana, abstrakt, věnování, vložené grafiky bez vazby ke konkrétnímu článku a veškeré ostatní autonomní texty podobné povahy, pokud je není vhodné rozčlenit podrobněji.

Stejně jako *svazky* tvoří hierarchickou strukturu nad stránkami, nad *základními články* lze vybudovat hierarchickou strukturu tvořenou *články*. Ta však není svázána tak pevnými pravidly, je tedy např. možné (byť ne obvyklé) vybudovat ze základních článků dvě nezávislé hierarchie, členěné dle různých kritérií.

Definice 5 *Logicky souvislá uspořádaná množina* základních článků *tvoří* článek. Základní článek *je* článek.

Článek může být např. soubor všech dílů článků vyšlého na pokračování, soubor článků sdílejících stejné téma, či soubor grafik uveřejněných v dokumentu.

Za články se také považují všechny základní články. Narozdíl od stran a svazků, které se svojí povahou liší, základní články a články lze popsat stejnou množinou vlastností. O tom, co je základní článek rozhoduje nerozhoduje (narozdíl od strany) objektivní kritérium (být natištěn na jednom listu papíru), nýbrž pouze podrobnost, s jakou chceme popsat obsah a strukturu dokumentu. Pokud bychom chtěli dokument členit podrobněji, základní články bychom dále rozčlenili. Tyto části by se staly „novými“ základními články, přičemž původní rozčleněné základní články by se staly „pouze“ články.

Košatost logického členění velmi záleží na typu dokumentu: u odborných časopisů složených z většinou navzájem nezávislých kratších textů je tato struktura zpravidla velmi plochá, naopak např. u doktorské práce může mít logická struktura několik úrovní a složitou strukturu, zatímco fyzická struktura bude velmi jednoduchá.

Na fyzické členění lze pohlížet jako na speciální (privilegovaný) typ logického členění. Proto u dokumentů, které mají lineární strukturu (např. knihy), se logické a fyzické členění může překrývat. V takovém případě není vždy nutné budovat vedle fyzického členění paralelní shodné logické členění.

2.2.2 Rysy běžným dokumentům společné

V této kapitole se budeme věnovat vlastnostem *svazků* a *článků*, společným pro všechny běžné typy dokumentů – v další kapitole se pak budeme věnovat specifikům jednotlivých typů dokumentů.

Svazky

Klasické zaměření knihovních systémů je na zachycení fyzické struktury dokumentů. Projekt *Depozitum* se zaměřuje na vyhledávání informací z digitalizovaných dokumentů – proto se zaměřuje především na logickou strukturu dokumentů. Jelikož digitalizace a detailní zpracování dokumentů je časově a finančně náročná a není zatím jasné, jak rychle bude digitalizace pokračovat (respektive jak velké množství zdrojů se pro tento účel podaří získat), nepředpokládá se v první fázi projektu digitalizace takového počtu svazků, aby si to vynutilo potřebu vybudovat knihovní systém v tradičním slova smyslu.

Proto se ani tato práce nezaměřuje na podchycení informací o jednotlivých svazcích, nýbrž hlavně o podchycení informací o jejich jednotlivých částech (např. u časopisu o článcích). O samotných svazích se tedy zatím uchovávají pouze ty informace, které jsou relevantní pro vyhledávání v jejich částech, tzn. názvy, struktura, členění, datum vydání, číslování stran a autoři jednotlivých části.

V případě růstu projektu se předpokládá implementace některého ze standardních knihovních formátů. Výběr formátu se bude řídit pravděpodobně dle partnerů, se kterým bude projekt spolupracovat (viz kapitola 2.1), projekt je navržen tak, aby mohl v zmíněných formátech (s přihlédnutím k jejich specifikům a omezením) s okolím komunikovat.

Strany

Většina digitalizovaných dokumentů má strany číslované. Číslování stran je pro člověka přirozený způsob orientace v dokumentu, čísla stran jsou užívána v citacích a referencích. Proto je užitečné tuto informaci zachovávat. V neposlední řadě zachování čísel stran zjednodušují práci s digitalizovanými daty, neboť číslo strany lze využít jako (snadno kontrolovatelný) identifikátor dané strany.

Číslování stran nemusí být zcela pravidelné. Na některých stranách číslo schází (je však odvoditelné z čísel okolních stran), některé strany číslo nemají a ani jim nelze žádné (celé) číslo přiřadit, aniž by se porušila vzestupnost posloupnosti čísel stran (např. strany s obrázkem uprostřed časopisu, nebo několik stran ze začátku dokumentu, které by musely mít záporné číslo strany).

Vzhledem k tomu, že některé dokumenty se sestávají z více svazků, nemusí být číslo strany jednoznačným identifikátorem v daném dokumentu.

Definice 6 *Strany jsou v rámci daného* svazku *číslovány* průběžně*, pokud čísla stran, které mají natištěné číslo, tvoří vzrůstající posloupnost (při zanedbání tiskových chyb). V opačném případě jsou strany v svazku číslovány* separátně *– v tomto případě mají části tohoto svazku zpravidla vlastní číslování stran, začínající vždy od začátku.*

Klasickým případem průběžného číslování jsou časopisy, kdy první číslo z každého ročníku je číslováno od jedničky, zatímco druhé a další číslo svým číslováním navazuje na předchozí číslo. Naopak v případě separátního číslování by každé číslo začínalo stranou jedna. Je také možné, že v jednom svazku se vyskytuje více oddílů se separátním číslováním – v tomto případě lze však na takový svazek pohlížet jako na více svazků.

Strany, články a textová podoba Přestože je v projektu *Depozitum* kladen největší důraz na články, pro cíle, které si projekt *Depozitum* klade, není nutné rozdělovat

digitální kopie stran na části náležící jednotlivým článkům: zatímco dělení strany na jednotlivé kusy by bylo poměrně náročné, uživatel snadno na dané straně lokalizuje začátek či zaznamená konec článku.

Naopak u textové podoby má toto rozdělení textu strany k jednotlivým článkům poměrně velký význam: protože bez tohoto rozdělení je nutno přiřadit článku text z celé strany, byť z ní článek zabírá jen část – a fulltextové vyhledávání tak může vracet „falešné“ výsledky. Proto zatímco při práci s obrazovými daty bude nejmenší „adresovatelnou“ jednotkou strana, textová data budou členěny i na drobnější části.

Články

U článku je především třeba evidovat jeho umístění v dokumentu. Dle definic 4 a 5 lze každý článek mimo základní popsat rozsahem jemu podřízených článků. Základní článek by k přesnému popsání potřeboval udávat přesné oblasti jednotlivých stránek, na kterých se nalézá. Protože takovýto popis by byl velmi náročný, bude projekt *Depozitum* uchovávat pouze rozsah stran, na kterých je základní článek obsažen, a případné pořadí základního článku na stránce, v jakém byl vytvořen.

U článků je možno uchovávat následující atributy:

Klíčová slova a vlastnosti U článků je možné evidovat *klíčová slova*, které pomohou při vyhledání článků psaných na dané téma. Klíčová slova je vhodné *hierarchizovat*:

Definice 7 Hierarchizace *klíčových slov je podchycení vztahu „A je speciální případ B“ či „Význam A je obecnější případ významu B“ mezi dvěma klíčovými slovy.*

Provedením hierarchizace klíčových slov lze např. definovat, že články označené klíčovým slovem *neurologie* automaticky spadají pod klíčové slovo *medicína*. Při vyhledávání článků s klíčovým slovem *medicína* by pak systém měl zobrazit i články s klíčovým slovem *neurologie* (naopak nikoli). Takto hierarchizované jsou i běžné slovníky klíčových slov (LCC, UDC, DDC)[12] [35, 36, 23].

Článkům je také možno přiřazovat vlastnosti (např. citovanost daného článku apod.). S těmi lze zacházet jako s klíčovými slovy s přiřazenou hodnotou.

Autoři Dále je pro *články* možno evidovat *autora*. Některé články jsou však podepsány pouze zkratkou či iniciály, jedna osoba se často podepisuje různě – identifikovat pod daným podpisem konkrétní je tedy obecně netriviální úkol pro vědeckého pracovníka. V některých případech může být vhodné přiřazovat autorům klíčová slova či další vlastnosti.

U podpisu pod článkem bývá často uvedená i titul či funkce dané osoby. Všechny tyto informace je třeba zachovat.

Forma uchování digitalizovaných dat

Digitalizovaný dokument je primárně uchováván jako soubor obsahující digitální obrázek (respektive jako množina souborů jednotlivých stran, které obsahují digitální obrázek. U článků lze však také uchovávat textovou podobu článku, ta může sloužit k fulltextovému vyhledávání, nabízet lepší čitelnost, či sloužit k dalšímu využití při strojovém zpracování.

[12] Jde o slovníky pro takzvané desetinné třídění [12, 6].

2.2.3 Specifika jednotlivých typů dokumentů

Nyní se budeme věnovat jednotlivým typům digitalizovaných dokumentů, jejich běžnému fyzickému a logickému členění a jejich případným specifikům.

Časopis Časopis vycházel po číslech, která se dále sdružují do ročníků. Některé ročníky časopisů mají v každém ročníku zvláštní přílohy (např. v *Časopisu katolického duchovenstva* zvané *Slavorum*) – ty vycházely buďto vcelku s některým číslem, nebo po částech v každém čísle. Některé časopisy mají tolik ročníků, že se pro uživatelskou přívětivost vyplatí je dále členit – např. po desetiletích. Většina originálů digitalizovaných časopisů je ve formě svázaného ročníku.

Každé číslo časopisu lze *logicky* členit na *články* – souvislé stati na jedno téma vymezené nadpisem a nadpisem dalšího článku. Delší stati vycházely často na pokračování v různých (většinou, ne však nutně, po sobě jdoucích) číslech, není výjimkou přesah takovéhoto seriálu přes hranice ročníku. Není vyloučeno ani pokračování článku v tom samém čísle. U některých časopisů je k dispozici i obsah ročníku – seznam publikovaných článků s udaným rozsahem stran v kvalitě vhodné pro užití OCR.[13]

Strany časopisu bývají číslovány průběžně v rámci ročníku či separátně. Většina přílohy má strany číslovány taktéž separátně. U časopisu mají smysl všechna metadata uvedená v kapitole 2.2.2 na straně 14.

Soupis památek Soupis památek je historická kniha z devatenáctého století obsahující krátké statě o historických památkách na území Čech a Moravy. Je členěna po kapitolách odpovídajících jednotlivým okresům; každé kapitole je přiřazeno číslo. Navíc má několik nečíslovaných kapitol o pražských památkách. Každý okres má vlastní číslování stran. Tato kniha lze logicky členit na stati o jednotlivých památkách.

Slovník Slovník je členěn po heslech (stať definující jedno slovo), strany má číslované jako běžná kniha. Slovník je až na výjimky dané pravděpodobně typografickou úpravou či tiskovou chybou seřazen dle abecedy. Žádné heslo nezasahuje do více stránek. Fyzicky lze slovník členit dle oddělení obsahující hesla začínající na jedno písmeno, logicky po jednotlivých heslech. V současné době započínají práce na digitalizaci historického latinského slovníku.

U tohoto a předchozího typu dokumentů je na zvážení, zdali u nich má smysl evidovat klíčová slova a autory jednotlivých hesel. Zatímco např. u latinského slovníku tato informace není často příliš podstatná (navíc, je-li psán jednou osobou, lze tuto činnost provést automaticky skriptem), např. u výkladového slovníku či encyklopedie to může být velmi podstatný údaj.

(Převážně historická) vydání Bible Bible se fyzicky člení na Starý a Nový zákon, sestávající se z několika desítek knih. Každá kniha se sestává z kapitol, které lze považovat za základní články. U Bible je sice definováno autorství, není ho však třeba zadávat – jednodušeji to lze udělat až hromadně pro všechny články (kapitoly) dané knihy Bilbe.

Jelikož majorita digitalizovaných dokumentů jsou časopisy a zároveň jsou to z digitalizovaných dokumentů dokumenty nejkomplexnější, dále budeme analyzovat hlavně je, přičemž zmíníme případné odlišnosti u ostatních typů dokumentů.

[13] `http://en.wikipedia.org/wiki/Optical_character_recognition`

2.3 Požadavky na systém digitalizace

Před návrhem systému je třeba shromáždit požadavky na systém – a to jak požadavky, které jsou dány možnostmi zadavatele, tak i požadavky uživatelů, který budou tento systém užívat.

2.4 Ekonomomičnost vývoje a priority projektu

Jak již bylo nastíněno v úvodu této práce – v současné fázi projektu, kdy se ověřuje životaschopnost a perspektivy projektu, testují se možné postupy a nejvhodnější řešení a hledá se vhodný ekonomický model projektu. Tomu jsou také podřízeny priority projektu: požadavkem zadavatele (Katolické teologická fakulta) není zcela hotová a bez zásahu programátora funkční aplikace pro digitalizaci dokumentů všech možných typů – taková aplikace by byla nad finanční i časové možnosti zadavatele. Pro prezentaci výsledků, získání případných dalších grantů a spolupracujících subjektů na projektu je také třeba v relativně krátkém časovém horizontu. Proto jsou při vývoji klíčové následující priority:

- Předpokládá se další spolupráce autora na projektu (popřípadě jiného programátora), proto pro administrační úlohy, které se provádějí zřídka nemusí existovat „laické“ uživatelské rozhraní. Tyto úkony však musí být kvalitně dokumentovány.
- Prioritou je rychlé pořízení kvalitních prezentovatelných výsledků.
- Během implementace i užívání projektu je nutné očekávat změny zadání, rozšiřování projektu apod. V navrženém systému musí být tyto změny dobře implementovatelné.

2.4.1 Distribuovanost systému

Jelikož na digitalizaci spolupracují i externí zaměstnanci fakulty (např. i studenti formou brigády), je třeba proces navrhnout pokud možno distribuovaně. Jednotlivé úkony v rámci procesu digitalizace by měli být rozdělitelné mezi více lidí a být na sobě co nejvíce nezávislé. Potřebné práce je také třeba rozdělit tak, aby jejich co největší část bylo možné vykonávat externě (např. prací z domova apod.).

Zároveň je třeba rozdělit potřebné úkony na takové činnosti, které může provádět nekvalifikovaná osoba (např. samotné scanování dokumentů) a na úkony vyžadující kvalifikaci (např. identifikaci autorů dle jejich pseudonymů vyžaduje vzdělání v daném oboru apod.). Přičemž by rozdělení mělo být takové, aby minimalizovalo práci, ke které je kvalifikovaná osoba potřeba. Podmínky uvedené v této kapitole vedou k minimalizaci nákladů na digitalizaci dokumentů.

Pro činnosti vykonávané nekvalifikovanými osobami je třeba navrhnout nástroje, pomocí kterých může vedoucí projektu kontrolovat rychlost práce jednotlivých pracovníků, či stanovovat výši ohodnocení jednotlivých uživatelů.

2.4.2 Ochrana dat

Při procesu digitalizace je pořizováno velké množství dat, což je spojeno s poměrně značnými náklady na jejich pořízení. Jelikož (v podstatě jediným) výstupem této práce jsou digitalizovaná data, je cena těchto dat značná; a to i pomineme-li jejich hodnotu z hlediska autorských práv či hodnoty historických informací a budeme brát v úvahu pouze náklady vynaložené na jejich pořízení.

Proto je třeba stanovit pravidla pro zacházení s digitalizovanými daty, které by zamezili jejich ztrátě, postup při jejich zálohování apod. Zároveň je třeba zamezit jednotlivým pracovníkům přístup k datům, na kterých nepracují tak, aby nemohli způsobit jejich poškození.

2.5 Uživatelé systému

Největší zájem o digitalizované materiály mají vědečtí pracovníci z různých univerzit – nyní digitalizované materiály spadají převážně do oblasti humanitní: společenské vědy, filozofie, teologie a historie, samotné dokumenty jako historický materiál se taktéž dají zařadit do oboru historie. Vzhledem k tomu, že některé materiály pochází z relativně nedávné minulosti (např. časopis *Via* apod.), dá se očekávat i zájem pamětníků o publikované dokumenty.

Vzhledem k převážně netechnickému vzdělání uživatelů systému, navíc jejich možnému vyššímu věku, by měl systém poskytovat jednoznačné, intuitivní a nesložité ovládací rozhraní. Naopak někteří vědečtí pracovníci, jejichž výzkum je zaměřen na oblasti, kterých se materiály týkají, budou službu užívat opakovaně – ti naopak ocení co nejefektivnější ovládací rozhraní, které (i na úkor jednoduchosti) umožní co nejrychlejší práci se systémem.

Vzhledem k tomu, že užívání služby bude vyžadovat dostatečně rychlé internetové připojení, dá se předpokládat u uživatelů přinejmenším průměrná vybavenost informačními technologiemi – aplikace pro zveřejnění digitalizovaných dokumentů tedy může spoléhat i na poměrně nové technologie (např. korektní implementaci *DOM*[43] a *ECMA skriptu*[8] (dále budeme používat „neoficiální“ leč zaběhnutý název *javascript*) odpovídající standardům *W3C* a *ECMA* v internetovém prohlížeči apod.).

2.5.1 Nejčastější činnosti uživatelů

Uživatelé digitálních knihoven mohou mít při studiu zdigitalizovaných historických knihovních fondů následující cíle:

1. Uživatel si chce dokument přečíst, popř. se pouze podívat, jak daný dokument vypadal, o čem pojednával apod. Tento cíl studia dokumentu mohou mít např. vědci studující dané historické období nebo laičtí zájemci o danou publikaci.
2. Uživatel dohledává citaci popř. její kontext uvedenou v jiném dokumentu. Ví tedy přesně jakou publikaci i jakou stranu chce nalézt. Je třeba mu dát k dispozici nástroje pro co nejrychlejší vyhledání požadované informace.
3. Uživatel ví, který článek si chce přečíst, neví však, v které publikaci popř. na jaké straně publikace je uveřejněn.

4. Uživatel provádí výzkum o nějaké osobě. Chce tedy vyhledat všechny články, které daná osoba publikovala.
5. Uživatel provádí výzkum na dané téma, či výzkum vývoje názorů na dané téma. Chce tedy vyhledat všechny dokumenty tomuto tématu se věnující.
6. Uživatel provádí lingvistický výzkum daného slova - chce tedy nalézt všechny výskyty tohoto slova.

Ovládací rozhraní programu by mělo reflektovat tyto přístupy k dokumentům a nabízet patřičné funkce.

Kapitola 3

Návrh projektu *Depozitum*, cíle a prostředky k jejich dosažení

V této kapitole se budeme věnovat cílům projektu *Depozitum*, prostředkům a postupům, které budou sloužit k dosažení těchto cílů a omezením, které je třeba při realizaci projektu brát v úvahu. Z této analýzy vyplyne návrh procesu digitalizace, tak i potřebné (převážně softwarové) nástroje, potřebné k provedení jednotlivých úkonů.

3.1 Cíle projektu

Projekt *Depozitum* dává důraz na umožnění co nejsnažšího využití informací v digitalizovaných dokumentech; snaží se tedy o detailnější anotaci daných dokumentů (až po úroveň *článků*), včetně užití klíčových slov i fulltextu.

Projekt *Depozitum* se snaží o obrácení zažitého paradigmatu o hledání informací: dosud při hledání konkrétní informace bylo nejprve nutno vyhledat knihovní jednotku (kniha, časopis...), ve kterém jsou informace uloženy, a teprve v ní informaci dohledat. Pokud bádající osoba nevěděla, v které knihovní jednotce se daná informace nalézá, bylo její nalezení velmi obtížné.

Cílem projektu *Depozitum* je umožnit při vědecké práci postup opačný: umožnit badateli nalézt potřebné informace, ať jsou již v kterékoli dokumentu – dle klíčových slov, fulltextového vyhledávání, jména autora apod. – a teprve bude-li třeba, i knihu, v které jsou tyto informace obsaženy.

Rozdíl mezi těmito přístupy je možno částečně přirovnat k postupnému vývoji v přístupu k navigaci a vyhledávání informací na internetu. Zatímco první generace vyhledávačů byla zaměřená na katalogy, striktně členěné dle některého kritéria (např. dle zaměření evidovaných firem) a s filosofií: nejprve hledej webové stránky věnující se tématu a na nich teprve nalezni to, co potřebuješ. Dnes jsou však daleko více využívané internetové vyhledávače: ty nejprve vyhledají přímo patřičnou informaci (kus textu věnující se danému tématu) a teprve pokud uživatel chce, má možnost od této „informace“ přejít na (celou) webovou stránku tuto informaci obsahující.

Narozdíl od klasických knihovních systémů, kdy je „základní jednotkou“ dokument a uživatel při jejich využívání je nucen nejprve patřičný dokument vyhledat, systém *Depozitum* se snaží co nejvíce zpřístupnit informace v daných dokumentech přímo. To vede k větší efektivitě práce badatele.

3.1.1 Přístup k informacím

Pro využitelnost digitalizovaných informací je tedy nejdůležitějším bodem snadná dostupnost a možnost snadného vyhledávání požadovaných informací. Dle analýzy požadavků uživatelů by projekt *Depozitum* měl nabízet následující přístupy k informacím:

1. Přístup ke knihovním jednotkám dle tradičních „knihovních“ metod (vyhledání dokumentu dle názvu, přístup k časopisu dle ročníku a čísla apod).
2. Přístup k jednotlivým statím vyhledaných dle názvů, autorů dle jejich podpisu (a případných dalších anotací) jednotlivých statí a vazbami mezi těmito staťmi.
3. Vyhledaným pomocí fulltextového vyhledávání.
4. Zpřístupnění textové podoby jednotlivých statí (např. článků časopisů)
5. Přístup k jednotlivým statím na základě přiřazených klíčových slov.
6. Vyhledání všech děl jednotlivých autorů, včetně autorů publikujících pod pseudonymy či s neúplným podpisem.

3.2 Prostředky k dosažení cílů

Cíle, zmíněné v předchozí kapitole, lze dosáhnout postupnými kroky, v kterých se z primárního zdroje: tištěného (popř. ručně psaného či malovaného) dokumentu získají patřičné informace, schopné (spolu s patřičným softwarovým nástrojem) splnit daný cíl. Tyto kroky na sebe sice navazují, lze je však snadno paralelizovat metodou pipeline (po dokončení prvního kroku digitalizace na jednom svazku (či dokonce straně) lze s tímto svazkem okamžitě zahájit krok druhý, i když zbylé svazky ještě nebyly vůbec zpracovány). Proto při dostatku prostředků může být digitalizace provedena v poměrně krátkém čase. Schéma celého procesu digitalizace včetně návaznosti jednotlivých procesů je na schématu 3.1, jednotlivým krokům se budeme věnovat v této kapitole.

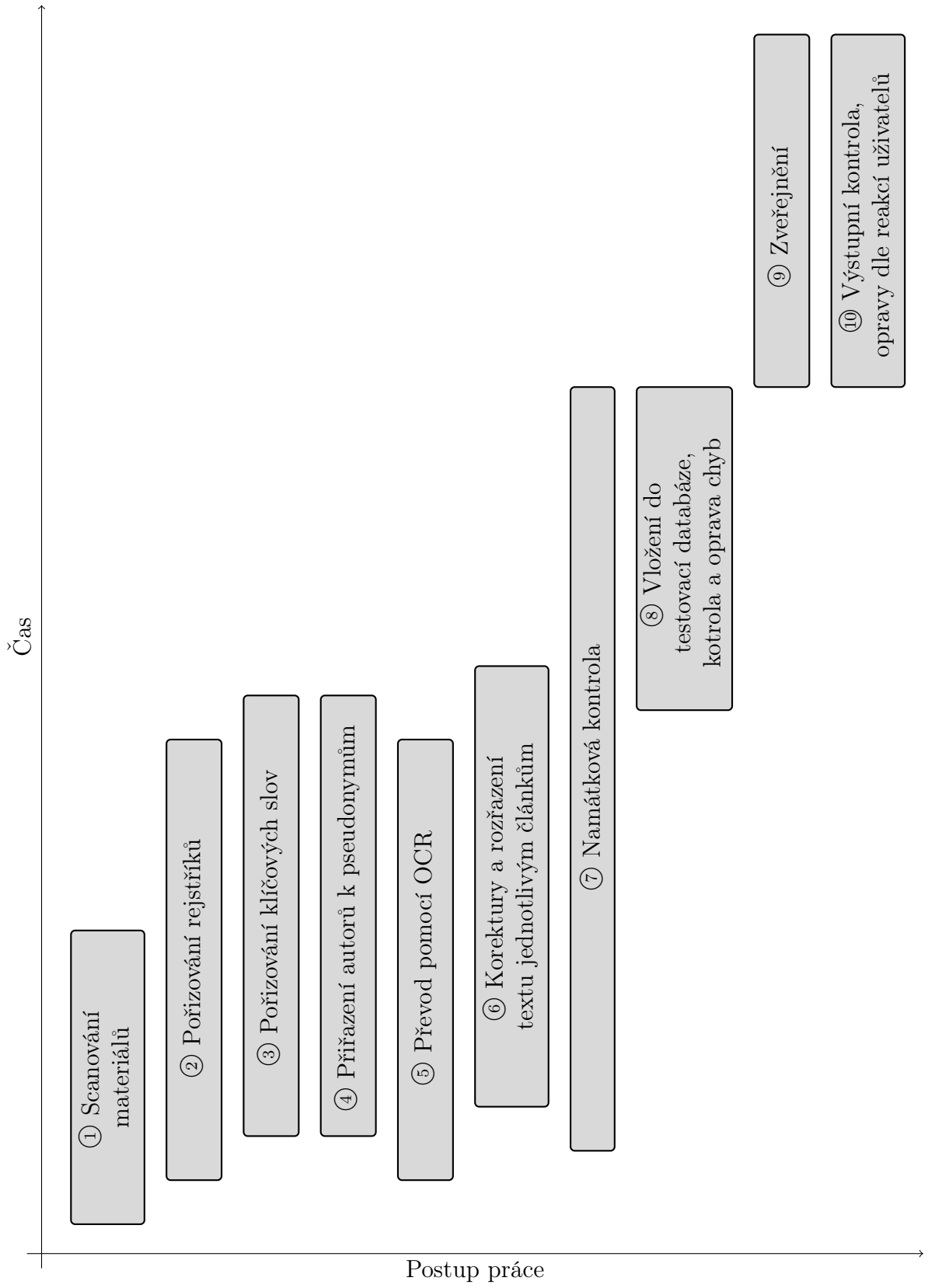
Přístup k digitalizovaným knihám... (cíl 1)

Prvním krokem, který je pro digitalizaci dokumentu třeba provést, je převod dokumentu do digitální (obrazové) podoby (scanování dokumentu). Při přípravě na scanování dokumentů je zároveň vhodné provést fyzické členění dokumentu na jednotlivé svazky; získaná data lze pak pojmenovat dle svazků, což zaručí snadnou orientaci v těchto datech bez nutnosti k souborům přiřazovat externí popis.

Provedením digitalizace a rozčleněním dokumentů do svazků bude získáno dostatek dat pro splnění cíle 1 z předchozí kapitoly.

Vytvoření rejstříku... (cíl 2)

Již zdigitalizované svazky je třeba rozčlenit na jednotlivé články, tedy vytvořit k dokumentu *rejstřík*. Tuto činnost lze provádět skoro paralelně s digitalizací: jelikož vytváření rejstříku trvá déle než samotné scanování, lze s pořizováním rejstříků začít již po převedení prvních stránek časopisu do digitální podoby.



Obrázek 3.1: Základní stručné schéma procesu digitalizace a časové návaznosti jednotlivých úkonů. (Nezahrnuje případné větší opravy chyb apod.).

Zpracování rejstříků vyžaduje ruční projití a zkatalogizování digitalizovaných tiskovin. Proto je tato činnost dosti časově náročná – byl tedy pro ni vyvinut pracovní postup a potřebné nástroje, které umožní pracovníkům tento rejstřík vytvářet dostatečně efektivně (při zachování ostatních požadavků na systém zmíněných v kapitole 2.4.1 na straně 18).

U některých časopisů jsou na konci ročníků vytištěné obsahy. Ty lze převést pomocí OCR a použít k automatickému generování rejstříků. Samotný převod pomocí OCR (jak dále ukážeme) však není úplně jednoduchý, i úprava převedeného textu do formátu, který lze dále automaticky zpracovávat, není zcela triviální záležitost. Navíc zpravidla vytištěné obsahy neobsahují tolik a tak kvalitních informací, jako lze získat přímo z dokumentu. Proto je třeba u každého dokumentu pečlivě zvážit, zdali je vhodné a ekonomicky výhodné tento způsob používat. [1]

Jelikož logické členění běžně digitalizovaných dokumentů je velmi ploché a lze ho postihnout návazností jednotlivých základních článků, je třeba zadávat pouze základní články. Po dokončení rejstříku se skriptem zpracuje seznam článků a pro seriály se vytvoří „zastřešující“ články.

Převod pomocí OCR... (cíl 3 a 4)

Převod nascanovaných dokumentů do textové podoby by vyřešil zároveň cíle 3 a 4. Bohužel tento na první pohled jednoduchý převod: aplikace OCR programu, v sobě skrývá velké obtíže. Digitalizované texty používají různých fontů - často však odlišných od fontů dnes standardně používaných. Některé texty v digitalizovaných dokumentech jsou vícejazyčné (včetně jazyků, které pro svůj zápis nepoužívají latinku, např. řečtina), starší texty navíc používají starou češtinu: ta má jak odlišný pravopis[2], tak i pravidla pro skloňování a časování, stejně jako tvary některých slov. To velmi stěžuje či vylučuje použití technologií pro rozpoznávání obrazových znaků OCR. Kvalita papíru a tisku některých dokumentů taktéž stěžuje aplikaci této technologii.

Byly provedeny orientační testy na užití programu OCR[3].Výsledná kvalita textu byla velmi různorodá: poměr správně rozpoznaných slov a všech slov velmi kolísal. U velmi poškozených dokumentů (velmi řídký případ) dosahoval cca 20% (většinou celé úseky textu byly nečitelné), u textů z devatenáctého století se starým pravopisem a cizojazyčnými pasážemi se pohyboval od 60% do 90% (Časopis Katolického Duchovenstva), u novějších, kvalitních podkladů z přelomu 19. a 20. století se současným pravopisem přesahoval 95% a často dosahoval až 99% (Časopis Nový život, ročník 1907).

Hlavním faktorem, který ovlivňuje výsledek rozpoznávání je kvalita tisku, dále pak kvalita samotného papíru a jeho čistota, velikost písma, typ pravopisu a počet užitých jazyků na stránce. Chyby vznikají často na místě, kde jsou buďto písmena či slova příliš blízko, je použit okrasný font, nebo není linie písmena souvislá (chyba tisku).

Tyto testy byly provedeny s programy, které rozpoznávaly pouze latinku. Pokud by programy měly zároveň rozpoznávat i jiné abecedy (hebrejskou, řeckou), kvalita by se pravděpodobně ještě více zhoršila.

U některých kvalitních předloh by tedy šlo uvažovat o využití automatického převodu do textové podoby, stále by však pro bezchybný text byly nutné korektury. U velké části

[1] V praxi se ukazuje, že u většiny dokumentů je ruční projití výhodnější.

[2] Např. dlouhé *í* se psalo jako *j*

[3] Testy byly provedeny pomocí software *ABBY Fine Reader 9.0* `http://www.abbyy.com/` a Recognita Omnipage 14 `http://www.nuance.com/`

dat by však po provedeném automatickém převodu byly nutné korektury; u starých tisků je pravděpodobné, že některé strany by bylo rychlejší přepsat ručně.

Data získaná automatickým převodem do textové podoby lze však využít k fulltextovému vyhledávání. Jelikož jsou články členěny drobněji než strany a z převodu pomocí OCR se získá text jednotlivých stran, musí být vyhledáváno vždy v textu celé strany. To způsobí, že fulltextové vyhledávání vrátí ve výsledku i nerelevantní články (ty, které leží na stejné straně jako vyhledávaný text). Naopak pokud nebudou opraveny chyby, nebude vyhledávání úplné. I přes tuto svoji nepřesnost a neúplnost je však textové vyhledávání pro uživatele přínosem. Ten však musí být na tento fakt upozorněn. Aby bylo textové vyhledávání přesné, je třeba text jednotlivých stran, rozdělit a přiřadit jednotlivým článkům.

Klíčová slova a anotace... (cíl 5)

Narozdíl od vytváření rejstříků je pořizování klíčových slov a přiřazování vlastností článkům netriviální úkol vyžadující od pracovníků vědeckou práci. Pracovníci, provádějící přiřazování klíčových slov a vlastností dokumentů, musí být v daném oboru vzdělaní, navíc tento proces požaduje zaškolení. Pokud nebude zaškolení či odbornost pracovníků dostatečná, dojde ke snížení relevance klíčových slov. Je možné uvažovat i o využití studentů daného oboru, o to pečlivější však musí být zaškolení a následná kontrola provedené práce.

Při přiřazování klíčových slov je také důležitá správná volba slovníku, z kterého se klíčová slova přiřazují. V současné době nedospěl (díky velké finanční náročnosti způsobené nutností kvalifikované práce) žádný z digitalizovaných dokumentů do fáze pořizování klíčových slov.

Před započetím přiřazování klíčových slov jednotlivým článkům je třeba rozhodnout, zdali se použije některý z užívaných standardů, např Dewey Decimal Classification, Uniwey Decimal Classification či The Library of Congress Classification [36, 23, 35], či jsou tyto standardy pro detailnější popis obsahu dokumentů nedostatečné a je třeba vytvořit slovník vlastní. V tomto případě by pro zachování zpětné kompatibility bylo vhodné některý z těchto standardů užít jako základ a vhodně ho rozšířit. Na tomto rozšíření by bylo vhodné spolupracovat s dalšími subjekty zabývajícími se katalogizací dokumentů a jejich obsahu. Toto rozhodnutí musí provést odborník na téma(ta) obsažené v digitalizovaném dokumentu, v součinnosti s odborníkem na knihovnictví.

Po převedení digitalizovaných dokumentů do textové podoby je možno uvažovat i o automatizovaném přiřazování klíčových slov pomocí specializovaného programu, který by provozoval automatickou klasifikaci textů. Podrobnější rozpracování tohoto tématu by však vydalo na samostatnou práci. Vzhledem k tomu, že samotný převod dokumentů do textové podoby je velmi náročný proces, nepředpokládá se v nejbližší době nutnost tento problém řešit.

Přiřazení pseudonymů k jednotlivým autorům... (cíl 6)

U knih je autorství (až na výjimky) jasné a předem dané. U časopisových článků jsou však autoři často podepsaní zkratkami, pseudonymy, či pouze příjmením.

Definice 8 *Pseudonymem budeme rozumět textový řetězec, kterým u článku určuje autora článku (plné jméno, příjmení, redakční zkratka, pseudonym...).*

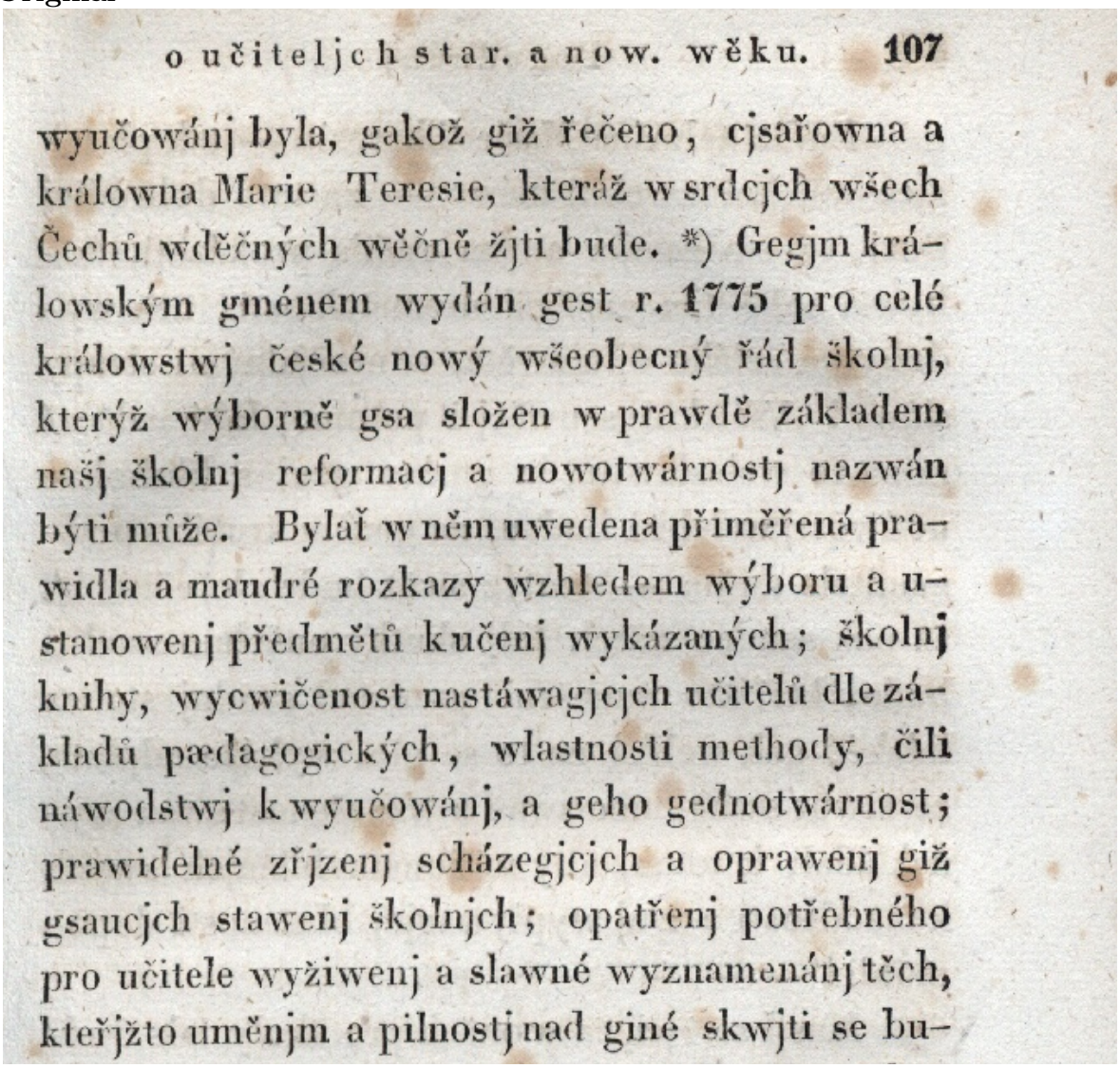
Je častým případem, že jeden autor používal více pseudonymů – proto je třeba přiřadit všechny autorské podpisy pod patřičnou osobu. Bude-li toto provedeno, bude v databázi dostatek informací pro splnění cíle 6.

Zpracování seznamu autorů je práce vyžadující opět kvalifikovanou osoby, která bude schopna odhalovat skutečnou identitu autorů, podepsaných pseudonymy. Jako předstupeň pro tento cíl lze u dokumentů, do kterých přispíval pouze omezený počet lidí, automaticky přiřadit všechny pseudonymy se stejným příjmením či stejnými iniciály pod jednu osobu. Tuto akci lze provést, i když se skutečné zpracování autorů (z ekonomických důvodů) neprovádí.

Dalším problémem spjatým s katalogizací autorů je možnost existence dvou osob stejného jména, či užívajících stejného pseudonymu, přispívajících do jednoho dokumentu (většinou časopisu). Tuto skutečnost nemůže nekvalifikovaný pracovník odhalit, proto je třeba, aby (kvalifikovaný) pracovník, který provádí přiřazování podpisů a pseudonymů skutečným osobám, zároveň prohlédl články u každého podpisu (či pseudonymu, zdali nenáleží více osobám, a případně rozřadil články správným autorům).

Pokud je k dispozici kvalifikovaná osoba, která má dobrou znalost daného dokumentu, není nutné kvůli duplicitám procházet všechny autory: stačí projít pouze ty, kde je známo že hrozí duplicita. V případě odborné publikace, kdy je okruh autorů limitován, postačuje i dobrá znalost daného oboru. Případně, pokud jde dané autory odlišit nějakým kritériem, které je schopna aplikovat i nekvalifikovaná osoba (např. různým časem působení v časopise), lze případné rozlišení svěřit i nekvalifikované osobě či autory rozlišovat již při zadávání rejstříků.

Originál



o učiteljch star. a now. věku. 107

wyučowánj byla, gakož giž řečeno, cjsařowna a
králowna Marie Teresie, kteráž w srdcjch wšech
Čechů wděčných wěčně žjti bude. *) Gegjm krá-
lowským gménem wydán gest r. 1775 pro celé
králowstwj české nowý wšeobecný řád školnj,
kterýž výborně gsa složen w prawdě základem
našj školnj reformacj a nowotwárnostj nazwán
býti může. Bylať w něm uwedena přiměřená pra-
widla a maudré rozkazy wzhledem výboru a u-
stanowenj předmětů kučenj wykázaných; školnj
knihy, wycwičenost nastáwagjcjch učitelů dle zá-
kladů pædagogických, wlastnosti methody, čili
náwodstwj k wyučowánj, a geho gednotwárnost;
prawidelné zřjzenj scházegjcjch a oprawenj giž
gsaucjch stawenj školnjch; opatřenj potřebného
pro učitele wyžiwenj a slawné wyznamenánj těch,
kteřjžto uměnjm a pilnostj nad giné skwjti se bu-

Rozpoznaný text

o učiteljch star. a *npw.* věku. 107

vvyiičowánj byla, gakož *gií* řečeno, cjsařowna a *knílowna* Marie Teresie, kteráž *wsrdcjch w.šech* Čechů *WdBČnýčh wčeně žiti* bude. *) *Gegjmkrá-lowským ginénein* wydán gest r. 1775 pro celé *kiálowstvj Bftftké* nowý *wSeobecný* řád skolnj, kterýž výborně gsa složen *wprawdS* základem našj skolnj reformacj a *nowotwSrnóstj* nazván býti *míiže. Bj Iai* w *uém* uwedena přiměřená pravidla a *mandié* rozkazy vzhledem *Wyboru* a *n— sianowcnj předmětn kučenj* vykázaných; skolnj knihy, *vycwičcnost* nastáwagjcjch učitelů dle *základ ú* pedagogických, *vlaslnosti meťhody*, čili náwodstwj k *wněowáiij*, a geho gednotwárnost; pravidelné zřjzenj scházegjcjch a oprawenj giž gsaucjch stawenj školnjch; opatřenj potřebného pro učitele *v)žiwenj* a slawné *wyzuamonánj* těch, *kii-í jžto* úměnjm a *piluosljnad* giné *skvjti* se bu-

Obrázek 3.2: Příklad kvalitou spíše podprůměrného historického podkladu a výsledku získaného OCR programem z Časopisu katolického duchovenstva. Chybně rozpoznaných slov (jsou vysázené kurzívou) je cca 35%.

Kapitola 4

Specifikace systému *Depozitum*

4.1 Proces digitalizace

V této kapitole se budeme věnovat jednotlivým krokům procesu digitalizace, prováděné v projektu *Depozitum*, Zatímco v předchozí kapitole 3.2 na straně 22 jsme se věnovali hlavně cílům digitalizace a potřebným činnostem (zachyceným na schématu 3.1 na straně 23) k jejich dosažení, v tomto oddíle práce se budeme věnovat jednotlivým nástrojům, které budou k digitalizaci sloužit a specifikaci pracovních postupům, které se při digitalizaci budou provádět.

Celý proces digitalizace lze rozdělit na čtyři hlavní části.

1. **Práce s primárními daty** Do této kategorie náleží digitalizace primárních dat a převod dat pomocí OCR. Pro tyto činnosti se budou užívat převážně externí nástroje, tato specifikace však stanoví pravidla a postupy pro digitalizaci.

2. **Práce s metadaty** Do této kategorie patří zadávání rejstříků, zpracovávání klíčových slov a autorů a přiřazování textu dokumentu jednotlivým článkům. V této části je třeba specifikovat aplikaci (popř. aplikace), pomocí které se budou dané činnosti provádět, a pracovní postupy pro užívání této aplikace. Tuto aplikaci budeme nazývat *Aplikace pro zadávání*.

3. **Zveřejnění dokumentů** Pro zveřejnění digitalizovaných dokumentů je také třeba aplikace. Tuto aplikaci budeme nazývat *Aplikace pro zveřejnění*.

Ve zbytku této kapitoly se budeme zabývat tématy, které nejdou zařadit do žádné z těchto kategorií, popř. činosti z těchto kategorií zastřešují. Ve zbylých dvou kapitolách se pak budeme věnovat pracovním postupům k dosažení vytyčených cílů, respektive specifikaci softwarových nástrojů, které bylo třeba pro tento účel vyvinout.

4.1.1 Řízení projektu

Celý projekt digitalizace řídí *vedoucí projektu*. Vedoucí projektu koordinuje jednotlivé práce, zadává podřízeným osobám práci, tuto práci přebírá zpět a kontroluje. Má k dispozici nástroje na sledování výkonnosti jednotlivých pracovníků. Vedoucímu projektu náleží některá rozhodnutí v procesu digitalizace (např. digitalizace v nadstandardní obrazové kvalitě, způsob rozdělení daného dokumentu na části apod.). Vedoucí také provádí procesy publikace digitalizovaných výsledků, rozhoduje o nasazení nových verzí aplikace

a provádí některé administrační úkony. Při růstu projektu má vedoucí možnost delegovat svoje pravomoci na další osoby.

Proces digitalizace se tedy sestává z několika, na sobě (v podstatě) nezávislých kroků. Každý z těchto kroků je však poměrně nákladný, navíc některé vyžadují práci kvalifikovaných a/nebo zaškolených osob. Vedoucí projektu proto musí u každého digitalizovaného dokumentu (dle dostupných financních prostředků a dostatku pracovních sil) zvolit, zda a kdy se bude provádět která fáze digitalizace. U některých činností (zpracování autorů, převod do textové podoby) musí vedoucí zvolit mezi úplným manuálním zpracování, které je finančně značně nákladné, a využitím automatizace, která však obvykle přináší o mnoho horší výsledky.

4.1.2 Unifikace různých typů dokumentů a terminologie

Jelikož do digitalizace vstupují dokumenty různých povah, je třeba stanovit pro celý proces digitalizace a publikaci digitalizovaných dokumentů taková pravidla, která zároveň umožní respektovat při digitalizaci rozdílné povahy jednotlivých typů dokumentů. Pokud by však pro různé typy dokumentů byla stanovena různá pravidla, vedlo by to jednak k větším nákladům při vývoji systému (pro každý typ dokumentů by se musel vyvíjet paralelní systém), jednak to přináší větší riziko chyby (aplikace nesprávné sady pravidel na daný dokument).

Proto si stanovíme následující termíny, které umožňují zacházet s díly soupisu památek, respektive s oddílem slov začínajících jedním písmenem u slovníku a dalšími typy dokumentů a jejich částí stejně jako s ročníkem časopisu (s několika výjimkami, např. neexistence autorů apod.). Termíny jsou definovány obecně, což umožňuje snadnou aplikaci pravidel i pro další možné typy dokumentů.

Definice 9 Díl svazku *je* svazkek*, která do tohoto* svazku *náleží (např. číslo je díl ročníku či časopisu, ročník je díl časopisu apod.).*

Titul *je* svazek*, který není částí jiné* svazek.

Číslo *je* svazek*, který neobsahuje žádné (další)* svazky.

Ročník *Přímá část* titulu *– tedy neexistuje* svazek*, který by nebyl* titul *a tento* svazek *by byl jeho částí (tzn. např. ročník časopisu, nebo soupis památek jednoho okresu, nikoli však číslo časopisu).*

Příloha *je* svazek*, jehož obsah či/a způsob vydání se odlišuje od standardního obsahu* svazků *daného* titulu*. Vycházela-li příloha po částech v jiných* svazcích*, je možné buďto považovat všechny tyto části* svazku *za jeden* svazek*, či/a každý díl přílohy za samostatné* svazky.

Seriál *Články, logicky k sobě náležející v rámci jednoho* titulu*, se nazývají* Seriál.

Jelikož každý dokument má svá specifika a nelze předvídat, jaké všechny typy dokumentů budou digitalizovány, nejsou zde definované termíny definované vždy přesně (např. zdali daná část dokumentu je nebo není přílohou, zdali se má daný dokument či jeho část při zpracování rozdělit na více *svazků* či nikoli apod.). O vhodné aplikaci těchto definic rozhodne vedoucí projektu.

4.1.3 Zálohování dat

Nejcennějším výsledkem projektu *Depozitum* jsou získaná data. Proto musí být patřičně chráněna. V celém systému *Depozitum* je zajištěno, že všechna data se vyskytují minimálně na třech nezávislých místech. Primární úložiště s daty je pravidelně zálohováno, navíc jsou pořízená data (vždy když je dokončen úsek práce) zálohována na externí média, která nejsou připojena do elektrické ani datové sítě – tím je zamezeno ztrátě dat i při přepětí v celé rozvodné síti či totálnímu selhání zabezpečení intranetu.

Pro opravdu spolehlivé uchování dat je třeba počítat i s mimořádnými událostmi, jako je živelná katastrofa, válka apod. Proto by alespoň jedna záloha měla být fyzicky umístěna na geograficky zcela odlišném místě než ostatní data. Ideálním řešením (z pohledu vynaložených nákladů) se jeví navázat spolupráci s některou z univerzit či jinou vědeckou institucí (nejlépe na jiném kontinentu) zabývající se digitalizací a vzájemně si poskytovat prostor pro zálohování.

Dalším nebezpečím pro uchování dat může být zvolený typ záznamových technologií. Některé technologie nemají příliš dlouhou trvanlinost (např. vypalované CD či DVD), a tak nejsou pro uchování dlouhodobých záloh příliš vhodné. U většiny datových médií je pro maximalizaci životnosti třeba je uchovávat v patřičném prostředí (konstatní teplota, vlhkost atd.). Dnes nejrozšířenější způsob záznamu dat pomocí magnetického pole (ať již pevné disky či páskové mechaniky) je náchylný na poškození elektromagnetickým polem.

Proto by alespoň jedna kopie dat měla užívat odlišného způsobu záznamu než ostatní kopie (např. holografické paměti[1]). Jelikož tyto alternativní způsoby uložení dat jsou velmi nákladné, je možné tento požadavek nahradit speciální ochranou datových nosičů (např. uložit pevné disky obsahující zálohy do prostoru, dostatečně spolehlivě stíněného před elektromagnetickým polem).

Pokud mají zálohy sloužit k uchování dat i pro případ katastrofy, je nutné udržovat zálohy v takovém stavu, aby byly čitelné i po velmi dlouhé době, popř. po nepředvídatelné události (globální živelné katastrofě, válce). Proto je nutné zálohy pravidelně kontrolovat, zdali nejsou poškozené a v případě potřeby vadné nosiče nahrazovat novými. Dále je potřeba zajistit prostředky pro jejich přečtení: spolu se zálohami by tak měl být uložen i nástroj či pro jejich přečtení (např. v případě uložení záloh na pevné disky počítač, schopný s těmito disky komunikovat, v případě užití holografických pamětí navíc čtečka těchto pamětí).

Konkrétní řešení zálohování dat je rozebráno v kapitole 5.1.3 na straně 50.

4.2 Digitalizace (scanování) primárních dat

Prvním krokem pro publikaci dokumentů na internetu je jejich scanování: tedy získání digitálního obrazu jednotlivých stran dokumentu. Před započetím digitalizace rozhodne vedoucí projektu o rozdělení daného dokumentu na svazky a přidělení svazků jednotlivým pracovníkům ke scanování.

Pracovníci pořizují digitální kopie stran jim přiřazených svazků a ukládají je na lokální disk do adresářů dle ročníků. V dohodnutém intervalu pak výsledek své práce odevzdávají vedoucímu projektu, který data přesune do datového úložiště.

[1] např. `http://www.inphase-tech.com/products/media.asp?subn=3_2`

4.2.1 Hardware pro scanování dokumentů

Vytváření digitálních kopií se nazývá *scanování*, tento název je odvozen z (anglického) názvu pro počítačovou periferii zvané *scaner*, pomocí které lze digitální kopie vytvářet.

Pro scanování knih a podobných svázaných materiálů je vhodné užít specializovaných scanerů pro scanování knih: ty mají prosklenou plochu pro scanovanou dokumentu až do kraje. Díky tomu se nemusí kniha rozevírat do přímého úhlu a tak nedochází k ničení vazby knihy. Pro šetrnou digitalizaci zvláště cenných dokumentů lze užít i zařízení pro bezkontaktní digitalizaci – tato zařízení jsou však značně nákladná a tak se jejich užití předpokládá pouze u velmi cenných tisků, či při masové digitalizaci: rychlost těchto specializovaných zařízení je vyšší než rychlost klasických plochých scannerů a tak při velkém počtu digitalizovaných stran může být tato metoda ekonomicky výhodnější.

Nejvíce času při scanování dokumentů zabere příprava dané strany na scanování (např. otočení listu) a samotné fyzické snímání obrazu (digitalizace jedné strany trvá přibližně necelou sekundu, liší se dle typu scaneru a rozlišení). Proto by vyvinutí vlastního software pro scanování nemělo významný vliv na efektivitu práce při vytváření digitálních kopií. Jelikož tuto fázi digitalizace může provádět zcela nekvalifikovaná osoba, náklady na jeho vývoj by (alespoň v současné době) zdaleka převýšil ušetřené náklady zvýšením efektivity práce. Ke scanování se tedy používá dávkový režim běžného software dodávaný ke scaneru, popř. u typů s nedostatečným ovládacím rozhraní lze použít možnosti dákové scanování poskytované programy pro práci s grafikou (např. IrfanView).

Pokud by byla prováděna digitalizace velkého počtu dokumentů, je možné uvažovat o vývoji takového software, ten by mohl digitalizované strany automaticky ukládat pod správným jménem do správných adresářů a zároveň k těmto souborům připojovat informace o osobě provádějící digitalizaci, sledovat výkon jednotlivých pracovníků apod. Zatím se však o vývoji tohoto software z výše zmíněných důvodů neuvažuje.

4.2.2 Pravidla pro scanování – formáty a názvy souborů

Strany dokumentů se standardně scanují v rozlišení 300*dpi*, barevně s barevnou hloubkou 8 bitů na barvu a ukládají ve formátu užívajícího bezeztrátové komprese do souborů formátu *TIFF*.[2] Jako alternativní formát, umožňující uložit digitalizovaná data v požadované kvalitě byl zvažován formát *PNG*, formát *TIFF* byl vybrán proto, že je toto kódování mezinárodním standardem (*ANSI* i *ISO*) a zároveň je defakto standardem pro ukládání obrazů ve vysoké kvalitě.

Zvolené rozlišení je zcela dostatečné pro věrnou reprodukci tištěného originálu. Větší rozlišení či barevná hloubka by nebyla ani v delším časovém horizontu pravděpodobně využitelná, zároveň by neúměrně narostla jak potřebná kapacita pro uložení digitalizovaných dat, tak i čas potřebný k scanování každé strany. Soubory v menším rozlišení by naopak již neposkytovaly tak kvalitní reprodukci, čímž by se porušil jeden z cílů zachovat z digitalizovaných dokumentů maximum informací.[3]

V případě dokumentů vyžadujících větší obrazovou kvalitu (ručně psané předlohy, či tištěné předlohy s přípisy či s rytinami) je vhodné při digitalizaci užít větší rozlišení. Toto rozlišení by mělo být takové, aby každý nejmenší bod v originále nesoucí informaci (tedy např. tloušťka nejtenčího písma či vrypu rytiny) byl reprezentován nejméně dvěma

[2] `http://partners.adobe.com/public/developer/tiff/index.html`

[3] Pořizovaná kvalita např. převyšuje kvalitu, kterou používá *Národní digitální knihovna*, ta, přestože po partnerech požaduje minimální rozlišení 300*dpi*[3], sama užívá formátu *JPEG* a rozlišení 200*dpi*[4]).

obrazovými body v digitální kopii. Užité rozlišení pro daný dokument určí vedoucí projektu.

Název souboru strany

Každá digitalizovaná strana dokumentu je uložena v samostatném souboru. Dále budeme soubor obsahující digitalizovanou stranu nazývat *soubor strany*. Tyto soubory mají (vzhledem k souborům stran všech dokumentů) unikátní název souboru. V tomto názvu souboru je obsažen *kód svazku*, identifikujicí svazek, z kterého je strana pořízena, a číslo strany. Tento název je volen tak, aby zachovával pořadí stran v dokumentu. Výjimku z tohoto pravidla lze povolit u *příloh*, které vycházely postupně v jednotlivých číslech. Jelikož nemají logickou vazbu na předcházející a následující číslo, definice fyzického uspořádání je povoluje umístit na začátek či konec svazku, do kterého příloha náleží (např. u časopisu obvykle na začátku či konci ročníku).

Při scanování je třeba přiřadit jednotlivým souborům obsahujícím digitalizované strany název identifikující obsaženou stranu (viz kapitola 4.2.2 na straně 31). Pro definici tohoto formátu si zavedeme následující definici:

Definice 10 *Strana je* číslována v svazku *(jehož je strana součástí), v kterých tvoří čísla všech číslovaných stran (s výjimkou chyb v číslování) vzrůstající a logicky navazující posloupnost.*

Pro názvy těchto souborů byl tedy zvolen následující formát:

`<Identifikátor svazku>.<Číslo strany>.<Přípona>`

`<Identifikátor svazku>` *Identifikátor svazku* identifikuje svazek, v němž je strana daného titulu *číslována*.

Každý svazek může (ale nemusí) mít přiřazen *kód svazku* složený z alfanumerických znaků a podtržítka. *Identifikátor svazku* je složen z *kódů svazku* oddělených pomlčkou, které jsou po cestě od kořene fyzické hierarchie k danému svazku. Jeden kód i identifikátor může sdílet i více svazků. Svazky sdílející stejný kód používají společný „prostor“ pro číslování stran.

Kódy jsou svazkům přiřazeny tak, aby lexikografické uspořádání kódů zachovávalo přirozené pořadí svazků. Výjimka z tohoto pravidla platí pro přílohy, které mohou být zařazeny na konec či začátek nadřízené svazku. Každý titul by zpravidla měl mít přiřazen kód a měl by mít jednoznačný identifikátor.

Nechť časopis *Listy* se fyzicky na ročníky a čísla. Ročníky jsou číslovány *průběžně*: strany každého ročníku začínají stranou jedna, další číslo ročníku navazuje svým číslováním na číslování předchozího čísla ročníku. Ročník 1995 má dvě přílohy, jejicž číslování stran začíná opět od jedničky, jedna z příloh je však rozdělena do dvou svazků.

V tomto případě bude časopisu přiřazen kód `listy`. Každému ročníku je přiřazen kód rovný letopočtu. Čísla kód přiřazen nemají (neboť nevytvářejí separátní číslování, jejich strany jsou číslovány v rámci příslušného ročníku), ovšem přílohy ano. Ze tří svazků příloh ročníku 1995 budou mít dva svazky jedné přílohy přiřazen stejný kód `1` (neboť sdílejí číslování), druhá příloha

bude mít kód 2. Strany náležící do libovolného čísla tohoto ročníku budou mít identifikátor svazku `listy-1995`, strany náležící do první přílohy `listy-1995-1`, strany druhé přílohy `listy-1995-2`.

`<Číslo strany>` Číslo strany má formát `S[-M]`, kde `S` je nezáporné a `M` kladné celé číslo. Pokud má strana natištěné číslo, je `S` rovno tomuto číslu a `M` není uvedeno. V opačném případě je `S` stejné s předcházející stranou a `M` je o jedna větší než u předcházející strany.

Pokud natištěné číslo strany porušuje vzestupnost posloupnosti stran, je legální toto číslo opravit (např. je-li to evidentní tisková chyba, neboť předchozí strana má o dvě menší číslo než následující). Nelze-li chyba opravit, považuje se taková strana za stranu bez natištěného čísla. Pokud strana nemá natištěné číslo, ale toto číslo lze odvodit z čísel okolních stran (např. strana s celostránkovou reprodukcí obrazu, či strany ze začátku svazku, v které první číslovaná strana nemá číslo jedna, či absence čísla způsobené tiskovou chybou), je legální považovat takovouto stranu za stranu s natištěným číslem.

Pokud svazek začíná stranou bez natištěných čísel (s přihlédnutím k předchozím pravidlům), přiřadí se jí číslo `S-1`, kde `S` je o jedna menší než `S` první strany s uvedeným číslem. Číslo strany nemůže být záporné: pokud je v tomto případě v svazku přítomna strana s natištěným číslem nula, považuje se za stranu bez natištěného čísla a první strana svazku přebírá natištěné číslo strany nula.

Pokud je v některém svazku za sebou velké množství stran bez uvedeného čísla, lze tyto strany očíslovat, jako by měly číslo strany uvedené. Je-li třeba, je možné vydělit tuto část svazku jako samostatný svazek s přiděleným kódem.

Čísla `S` a `M` se uvádí vždy s tolika vedoucími nulami, aby všechny strany daného titulu měly řetězec `S[-M]` stejně dlouhý (tím se zajistí správné pořadí stran při procházení souborů stran seřazených dle jména).

`<Přípona>` identifikuje typ dat, obsažených v souboru. `tiff` značí formát tiff `jpg` značí formát *JPEG*, `djvu` formát *DjVu*. `txt`, `rtf`, `pdf` obsahuje text dané strany rozpoznaný pomocí technologie OCR ve formátu prostého textu, respektive formátu *rich text formát*, či *portable document formát*.

Způsob určování kódu dokumentům určí vedoucí projektu, zpravidla u ročníků a čísel časopisů je kódem letopočet či ročník respektive číslo daného čísla, u jiného díla by měl být kód vytvořen z názvu díla či jeho zkratky.

Vztah kódů titulů, *ISBN* a *ISSN*

Při přidělování kódu svazkům byla zvažována možnost užít standardní identifikátor knihy *ISBN*[13], popř. standardní idetifikátor periodika *ISSN*[14]. Standard těchto identifikátorů byl však vytvořen v roce 1966 respektive 1974 – tedy majorita v projektu digitalizovaných dokumentů nemá tento kód přidělen. Pokud je *ISSN* přiděleno, pak je přiděleno vždy celému titulu. Pokud je přiděleno *ISBN*, pak je přiděleno jednotlivým číslům.

V současné době vzhledem k počtu digitalizovaných dokumentů se kódy svazkům přidělují ručně. Ručně přidělené kódy vždy začínají písmenem. Při růstu počtu digitalizovaných dokumentů by se pro odlišení jednotlivých titulů a zajištění unikátnosti kódů

svazků by se kódy svazků měly volit kódy s čtyřciferným prefixem roku vzniku titulu a začínající písmenem (např. `1850ckd`). Pro tituly s přiřazeným *ISSN* je možné užít jako prefix osmiciferný kód *ISSN*.

Jelikož jednotlivá čísla užívají většinou průběžné číslování a tak nemají přiřazen vlastní kód, užití třinácticiferného *ISBN* jako kódu titulu připadá v úvahu pouze u digitalizovaných knih – nedá se však předpokládat, že se v rámci projektu bude digitalizován titul, který by měl přiřazen *ISBN*. Pokud by se tak stalo, je možné ho jako kód titulu užít.

Takto volený název souboru umožňuje snadnou kontrolu správnosti pořízených dat, nezávislost samotného scanování na dalších krocích digitalizace a snadnou identifikaci jednotlivých souborů bez nutnosti pořizovat k souborům zvlášstní soubor popisující jednotlivé soubory digitalizovaných stran. Soubory s digitalizovanými stranami jsou ukládány do datového úložiště.

4.2.3 Formát pro zobrazení na web

Po odevzdání souborů s digitalizovanými stranami dodatového úložiště vedoucímu projektu jsou z těchto dat vytvořeny kopie, sloužící k zobrazení na internetu a rovněž k další práci s materiály (např. pořizování rejstříků). Tím je zajištěna ochrana fyzického dokumentu, neboť s ním není nadále manipulováno. Tyto kopie budeme nazývat *kopie pro zveřejnění*. Soubory s kopiemi pro zveřejnění mají stejný název jako originální soubory až na odlišnou příponu identifikující typ souboru.

Kopie pro zveřejnění jsou obrazové soubory o délce delší hrany 1200px.[4] ve formátu *JPEG*[5] o kvalitě 60%. Tyto kopie jsou používány pro pořizování rejstříku a pro samotné publikování na internetu. Tento formát nabízí z běžně užívaných formátu na internetu (*JPEG*,*GIF*,*PNG*) pro zobrazení textu nejlepší poměr vizuální kvalita/velikost.

Byla testována i redukce obrazů do stupňů šedi. Velikost souboru se sníží pouze o cca 5%, přičemž visuální kvalita poklesne. Proto bylo od této redukce upuštěno.

Formát DjVu V současné době se uvažuje o užití formátu *DjVu*[6] pro zveřejnění dat. *DjVu* je optimalizován pro zobrazení digitalizovaných dokumentů a nabízí při podobné vizuální kvalitě cca poloviční velikost. Tento formát vyžaduje instalaci externího pluginu do internetového prohlížeče, navíc neumožňuje zobrazení obrázku přímo v internetové stránce, nýbrž pouze jako celého dokumentu (podobně jako např. zobrazení *PDF*). Tato nevýhoda však lze obejít pomocí *HTML* značky `IFRAME`.

Vzhledem k potřebě snadné prezentace projektu (z důvodu získání dalších finančních prostředků pro rozvoj projektu) je nutné zachovat formát *JPEG*. Zároveň se zdá rychlost stahování stran ve formátu *JPEG* dostatečná. Proto se v této fázi projektu zatím *DjVu* nepoužije, systém ale bude na nasazení tohoto formátu připraven. O nasazení formátu *DjVu* se rozhodne dle zpětné vazby od uživatelů – pokud by rychlost stahování obrazových dat nevyhovovala, bude implementována možnost volby mezi těmito formáty.

[4] V praxi docházelo k nedodržení přesného rozměru a u digitalizovaných dokumentů jsou tvořeny kopie o rozměru 1200px–1500px. Na funkčnost systému to nemá vliv a tak netrváme na znovuvytvoření kopií o „správné“ velikosti.

[5] http://www.jpeg.org/jpeg/index.html

[6] `http://djvu.org/`

Formáty *GIF* a *PNG* Při vybírání vhodného formátu jsme zvažovali i formáty *GIF* a *PNG*, které by teoreticky mohli nabízet větší ostrost textu. V provedených testech se však tyto formáty neosvědčily. Pro zachování stejné velikosti jako formát *JPG* v kvalitě 60% je třeba redukovat barevnou paletu na 8 až 16 barev (dle originálu): takováto barevná redukce je však již na první pohled zřetelná. Navíc v některých případech tato redukce zvýrazní kontrast vad papíru, což snižuje čitelnost textu. U nových časopisů, které používají čistě bílý papír a jsou tak teoreticky monochromatické by však bylo možné o jednom z těchto formátů uvažovat.

O správnosti zvolených formátů svědčí mimo jiné to, že i ostatní digitální knihovny používají taktéž formát *JPEG* či *DjVu*, některé knihovny užívají i obou formátů, např. *Národní digitální knihovna*.

4.3 Pořizování rejstříků a dalších metadat

Značná část práce při digitalizaci dokumentů je pořízení restříků a dalších metadat popisujících digitalizovaná data. Přestože tuto práci mohou vykonávat nekvalifikovaní pracovníci, je tento proces velmi pomalý a tedy i nákladný – pouze zaškolení a vycvičení pracovníci byli schopni zpracovávat strany podobnou rychlostí, jako byly scanovány.

4.3.1 Software pro pořizování rejstříků a dalších metadat

Pro pořizování rejstříků bylo voleno mezi několika možnostmi (užití excelových tabulek a následné zpracování, databáze v Accessu umožňující snadnou migraci práce aj.). Nakonec byla jako primární možnost pořizování rejstříků zvolena internetová aplikace.

Výhodou internetové aplikace je, že v primárním pracovišti na Katolické teologické fakulty je toto řešení svojí rychlostí v srovnatelné s rychlostí desktopové aplikace, lze ji však použít i mimo fakultu bez nutnosti instalovat libovolný software. V případě potřeby je možné její kopii zároveň poměrně snadno nainstalovat i na počítač bez možnosti připojení k internetu. Internetová aplikace umožňuje vedoucímu práce jednoduše kontrolovat postup prací, snadno opravovat již pořízená data i v zatím nedokončených a neodevzdaných rejstřících, snadnou zálohu (a případnou obnovu) dat, centralizovanou správu dat apod...

Největší výhoda internetové aplikace je však ve velmi snadné možnosti paralelní práce mnoha osob bez nutnosti slévat data: internetová aplikace zároveň umožňuje sdílet databázi autorů, klíčových slov a dalších společných zdrojů.

Automatické pořízení rejstříků Některé dokumenty (časopisy) mají v sobě uveřejněn obsah. Je-li tento obsah v dostatečné (obrazové kvalitě), je možné tento obsah převést pomocí OCR nástroje do textu, opravit chyby, zanést případné informace v obsahu chybějící a pomocí patřičného nástroje přenést do databáze. Pro tyto účely je vyvinut specializovaný skript na zpracování takto vygenerovaných obsahů.

Tento skript příjmá data ve formátu CSV (text oddělený středníkem) ve variantě Microsoft Excel. Formát je variabilní, první řádka definuje obsah jednotlivých sloupců tohoto souboru.

4.3.2 Uložení dat a jejich ochrana

Internetová aplikace (popř. zmíněný skript) ukládá data do datového úložiště (dále databáze). Jelikož pořízené rejstříky jsou nejcennější částí pořízených dat, musí být tato databáze pravidelně zálohovaná alespoň na dva rozdílné stroje, přičemž alespoň z jedné zálohy lze obnovit i historii databáze alespoň jeden měsíc nazpět (čímž se předejde nenávratné ztrátě dat při poškození databáze, které se zjistí až po provedení zálohy).

Přístup do internetové aplikace je chráněn uživatelským jménem a heslem. Existují dva druhy uživatelů: *běžný uživatel* a *administrátor*. Administrátor má přístup ke všem pořízeným datům a má právo zakládat, měnit a rušit uživatele, přiřazovat jim práva a kontrolovat postup práce. Běžný uživatel má práva měnit data týkající se svazků, které mu administrátor přiřadil, a jim náležících článků.

Každý svazek je přidělen maximálně jednomu uživateli, ten nese za data související s tímto svazkem zodpovědnost. Přidělením svazku se uživateli zároveň přidělí i všechny podřízené svazky přiděleného svazku. Data zadaná k svazkům, které uživatel nemá přiděleny, může běžný uživatel pouze prohlížet. Tato možnost je ponechána proto, aby pokud budou např. zvoleny nějaké konvence, jak daný rejstřík pořizovat (např. překlad cizojazyčného jména), mohl uživatel kontrolovat shodu v dodržování těchto konvencí s ostatními uživateli.

Knihovny

Každý digitalizovaný dokument je zařazen do jedné *knihovny* – všechny dokumenty z jedné knihovny sdílíjedno úložiště dat pro rejstíky a jsou na internetu prezentovány společně. O zařazení dokumentu do knihovny rozhoduje vedoucí projektu, v současné době se užívá pro každý titul samostatná knihovna, není však žádný problém zařadit do jedné knihovny více titulů.

Administrátor může omezit přístup uživatele pouze na jednu *knihovnu*. Uživatel s omezeným přístupem nejen nemůže měnit záznamy z ostatních knihoven, nýbrž ostatní knihovny ani nevidí (nejsou-li zveřejněny). Tato funkce slouží ke zpracovávání externích zakázek či k zpracovávání zakázek externími zaměstnanci.

4.3.3 Postup při pořizování rejstříků

Zpracovávání rejstříků probíhá obdodně jako scanování: vedoucí projektu rozdělí jednotlivým pracovníkům svazky (respektive u přiřazování pseudonymů autorům jednotlivé bloky nepřiřazených pseudonymů dle počátečního písmene) a zároveň jim předá kopie souborů stran (ve formátu pro zveřejnění). Pokud se tak již nestalo, zároveň vytvoří v databázi internetové aplikace záznamy pro jednotlivé svazky a přiřadí je patřičným pracovníkům.

Pracovníci procházejí přiřazené svazky stránku po stránce a postupně zapisují do databáze pomocí internetové aplikace záznamy o jednotlivých článcích, přiřazují článkům klíčová slova, či přiřazují pseudonymy k jednotlivým již existujícím autorům, popř. zakládají nové autory. Popíšeme data uchovávaná u běžného časopisu, případné odlišnosti u jiných typů dokumentů již byly zmíněny.

4.3.4 Pořizovaná data

V této kapitole popíšeme data, zadávaná pracovníky při pořizování rejstříků respektive dalších metadat.

Informace o svazcích O časopise se schraňují informace o jeho členění, tzn. jaké má ročníky a jaká čísla a případné přílohy spadají do daného ročníku. Pokud rozsah projektu bude růst, bude třeba uchovávat i další standardní „knihovnické“ údaje o daném svazku dle některého ze standardních knihovních formátů (viz kapitola 2.1 na straně 8). Svazky přiřazené patřičným pracovníkům zakládá vedoucí projektu, ti pak ve svých přiřazených svazcích mohou zakládat záznamy podřízených (vnořených) svazků.

Svazky jsou běžně řazeny dle alfanumerického pořadí (tzn. nejprve dle čísla, záznamy se stejným číslem dle porovnání řetězců). V případě potřeby lze záznamům přiřadit pořadí explicitně.

Informace o článcích – rejstřík U (základního) článku se eviduje do kterého svazku náleží, rozsah stránek v daném svazku, pseudonym(y) kterým je uložen autor článku a návaznosti mezi články. Návaznost mezi články se zadává výběrem předchozího a následujícího článku ze seznamu. Není-li návazný článek ještě v seznamu, vybere se speciální prvek seznamu „. . . není v databázi“. V seznamu pro výběr předchozího článku jsou nabídnuty jen články, které mají zadáno, že mají pokračování ale to ještě nebylo do databáze zadáno (a vice versa u následujícího článku). Výběrem předcházejícího/následujícího článku je modifikován i patřičný záznam u navazujícího článku.

U některých časopisů s cizojazyčnými či dnešní češtině příliš vzdálenými články je třeba evidovat i český název daného článku.

V standardních formátech (viz kapitola 2.1 na straně 8) je dále možnost evidovat u článku typ nadpisu, případně kategorii nadpisu (zdali je to opravdový nadpis, či pouze první řádka z textu apod.), či typ článku (recenze, polemika. . .). Katalogizace typu nadpisu nemá pro běžného uživatele příliš význam, proto bylo rozhodnuto jí neprovádět. Rozpoznání typu článku již požaduje alespoň zběžné prohlédnutí obsahu článku, navíc zaškolenou osobu schopnou rozeznávat typy článků. Takováto katalogizace má tedy spíše povahu přiřazení daného klíčového slova k článku identifikující jeho typ, než součást zcela „manuálního“ pořizování rejstříků – a je tedy zařazena mezi pořizování klíčových slov.

Články na jedné stránce jsou automaticky řazeny dle pořadí zadání. V případě potřeby je možné jim pořadí zadat explicitně.

Autoři Jelikož se autoři často podepisovali pseudonymy, u každého článku se eviduje *pseudonym*, který je rozdělen na část před příjmením, příjmení a část po příjmení. Tyto pseudonymy se zadávají při vytváření rejstříku článků, pseudonym lze buďto zadat, nebo vybrat ze seznamu již existující.

Tyto pseudonymy se následně přiřadí ke skutečným *osobám* (relace n:1). Toto přiřazení je nejprve provedeno skriptem: pod jednu osobu se sjednotí všechny pseudonymy dle příjmení. Poté je možno provést ruční revizi, přeřazení pseudonymů pod jiné osoby, úpravu automaticky vygenerovaných autorů, případné přiřazení klíčových slov aj. ve speciálním modulu aplikace.

Aby šlo kontrolovat, že žádný článek nebyl zadán do databáze, aniž by u něj byl zadán pseudonym, je zaveden speciální pseudonym *anonym*, který je „autorem“ všech

článků bez uvedeného autora.

Klíčová slova a anotace Pro definici klíčových slov je třeba nejprve definovat jejich slovník a hierarchizovat je. Následně je třeba projít všechny články a přiřadit kaaždému článku klíčová slova. Dále je možné článkům přiřazovat další vlastnosti: vzhledem teoreticky neomezené množině možných vlastností je tuto funkčnost vhodné implementovat jako slovník vlastnost – hodnota.

Textová data Pokud se na strany dokumentu použije *OCR* program na rozpoznání textu, je třeba nástroj na uložení rozpoznaného textu do databáze. V takto rozpoznaném textu mohou být provedeny korektury. Dále je vhodné rozdělit text stran na texty náležící k jednotlivým článkům. Systém je připraven na implementaci modulu aplikace, ve kterém by šlo tyto operace provádět.

4.4 Aplikace

Jak již bylo zmíněno, pro zprovoznění systému *Depozitum* byly vyvinuty dvě hlavní[7] aplikace: *„Aplikaci pro zadávání“*, sloužící k pořizování rejstříků a *„Aplikace pro zveřejnění“*, sloužící k zveřejnění digitalizovaných dokumentů. Obě tyto aplikace jsou internetové aplikace, přístupné přes webový prohlížeč. Kompatibilní jsou s *Internet Explorerem 6.0* a vyšším[8], *Mozilla Firefoxem*[9] (a dalšími prohlížeči založenými na jádře *Gecko*[10] a prohlížečem *Safari.*[11]

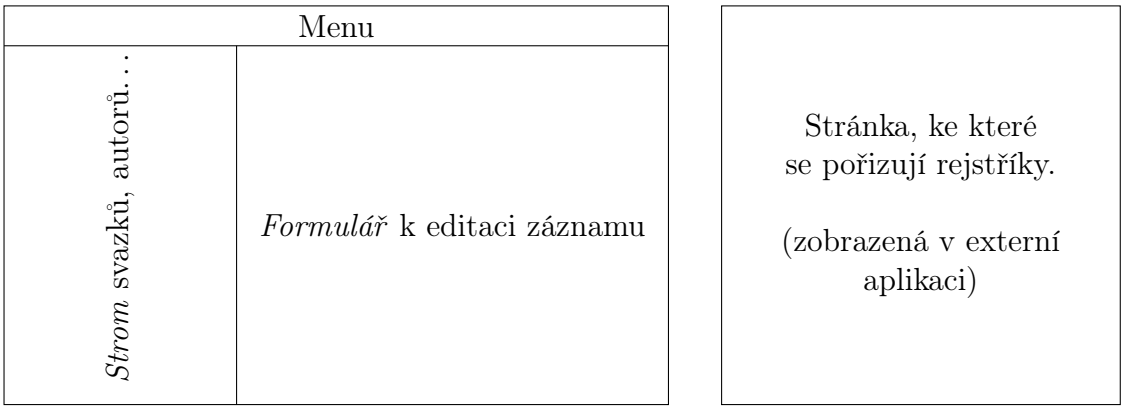
4.4.1 Aplikace pro zveřejnění

Aplikace pro pořizování rejstříků slouží pracovníkům provádějící digitalizace k pořizování rejstříků a metadat – pořizování obsahu dokumentů, přiřazování klíčových slov k článkům. . . . Tato aplikace je přístupná pouze oprávněným osobám.

Po přihlášení do aplikace se uživateli zobrazí okno prohlížeče rozdělené na tři části (viz schéma 4.1 na straně 39). V horní liště je menu, umožňující přístup k jednotlivým „modulům“ aplikace, odpovídajícím jednotlivým fázím digitalizace. Tyto moduly jsou následující:

- Pořizování rejstříků k článkům
- Správa autorů a jejich pseudonymů
- Pořizování klíčových slov a anotací k článkům
- Správa uživatelů
- Další nástroje a vyhledávání
- Nápověda k aplikacím

[7] Tzn. kromě dalších jednoúčelových skriptů a administračních nástrojů.
[8] `http://www.microsoft.com/cze/windows/products/winfamily/ie/default.mspx`
[9] `http://www.czilla.cz/produkty/firefox/`
[10] `http://wiki.mozilla.org/Gecko:Home_Page`
[11] `http://www.apple.com/safari/`



Obrázek 4.1: Základní vzhled GUI aplikace pro zadávání dat

Tyto moduly mají jednotný vzhled (výjimku tvoří moduly další nástroje a nápověda). Vlevo na stránce je *strom* (u správy uživatelů jednoúrovňový – tedy v podstatě list) editovaných záznamů, (svazků, autorů...). Tento strom slouží k výběru záznamu, který bude upravován v pravé (větší) části obrazovky. V té je umístěn *formulář* sloužící k editaci zvoleného záznamu.

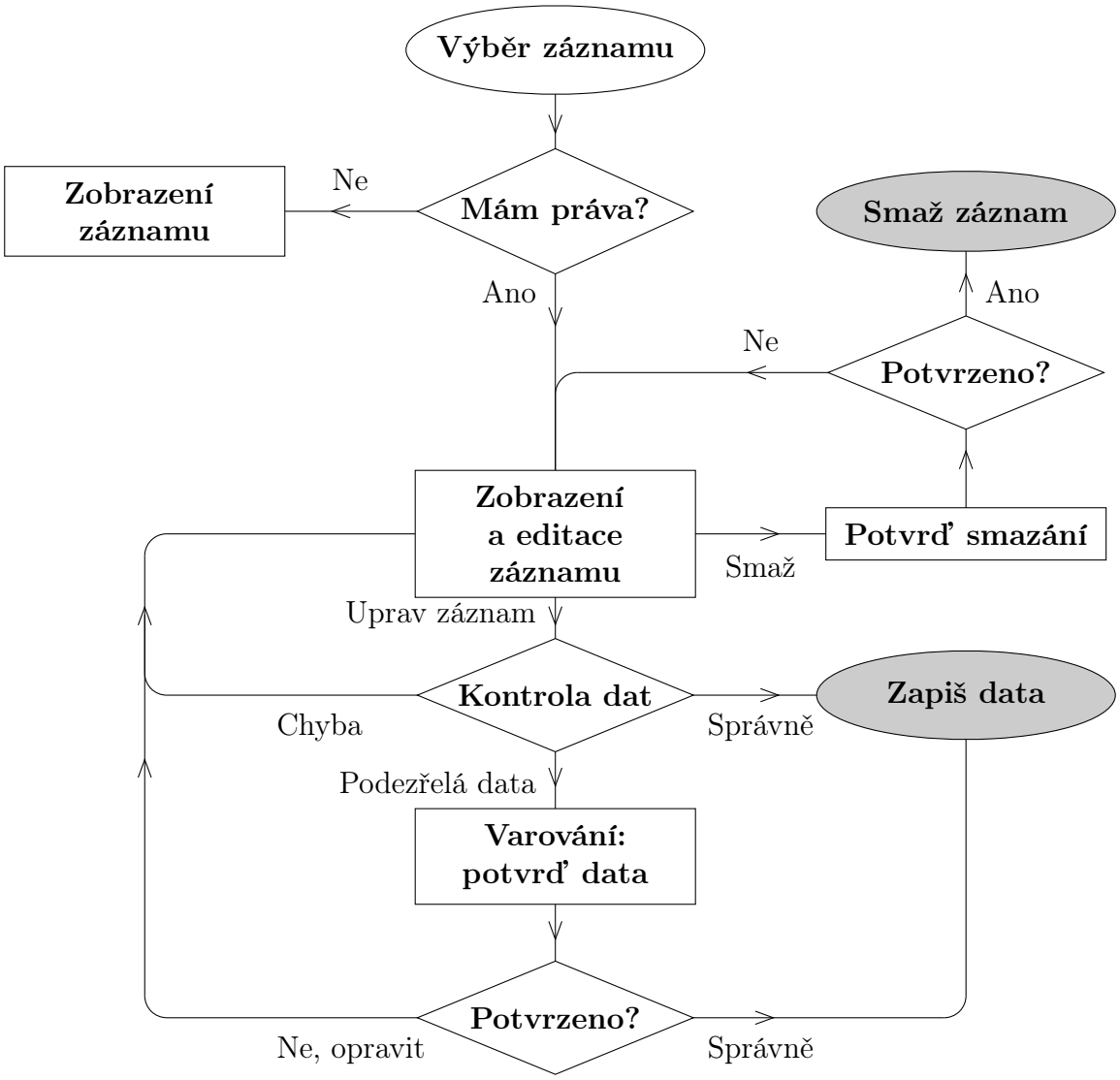
Vedle okna aplikace se předpokládá, že bude na monitoru otevřeno ještě jedno okno, obsahující prohlížeč digitalizovaných kopií stran, ke kterým pořizuje pracovník rejstřík, proto samotná aplikace neřeší zobrazení dat, ke kterým je rejstřík pořizován. Toto řešení je zvoleno proto, protože pracovník pořizující rejstříky obdrží (zmenšené na velikost pro snadné prohlížení) kopie digitalizovaných dokumentů, ke kterým pořizuje rejstřík (např. vypálené na CD). Tyto kopie si uloží na svůj lokální pracovní disk, ze kterého je načítá značně rychleji, než by je bylo možno načítat z internetu.

Uvažovalo se i o zabudování zobrazování těchto digitalizovaných stran do aplikace, ale od této myšlenky se upustilo z několika důvodů. Prvním a nejdůležitějším byl fakt, že užívaný způsob pořizování rejstříků zpracovávatelům zcela vyhovoval. Dalším důvodem jsou částečné technické obtíže: internetová aplikace z principu nemá přístup na pevný disk uživatele, takže názvy souborů stran by musely být přidány do internetové databáze již během procesu digitalizace, což zbytečně zesložiťuje proces digitalizace (čímž roste i riziko chyby). Navíc, jelikož se často vyskytuje více článků na jedné straně (a ze zadaného konce strany nelze poznat, zdali je tento článek na této straně poslední), musí si uživatel řídit listování stranami sám – integrace zobrazení stránek do aplikace by tedy nepřinesla zpracovateli výhodu. Proto byl ponechán původní „jednoduchý“ návrh.

Formuláře pro zadávání dat

Formulář pro editaci zvoleného záznamu může být ve třech módech:

- **Zobrazovací mód** Tento mód slouží k zobrazení dat (např. nemá-li uživatel k danému záznamu práva).
- **Editovací mód** V editovacím módu může uživatel měnit daný záznam.
- **Potvrzovací mód** Uživatel musí potvrdit svoji akci, pokud se snaží záznam smazat, nebo kdy je podezření na chybu ve vložených záznamech (možné chyby budou upřesněny u konkrétních typů záznamů). Záznam, u kterého je podezření



Obrázek 4.2: Schéma pro formulář k zadávání a úpravě dat

na chybná data budeme nazývat *podezřelý* záznam. k potvrzení akce slouží tento mód, kdy je zobrazena chtěná akce zároveň s ukládanou či aktuální (při mazání) podobou záznamu a možností akci potvrdit, či se vrátit zpět (k editaci či vytváření záznamu).

Diagram možných stavů formuláře je uveden na obrázku 4.2 na straně 40. Toto schéma platí pro případ úpravy záznamu, v případě vkládání nového záznamu není možné vstoupit do větve *smazání záznamu*.

Modul *Evidence svazků a článků*

V modulu pro evidenci svazků a článků jsou uzly stromu jednotlivé svazky, jako listy stromu jsou pak články těchto svazků. Vztah syn – rodič značí znamená syn (svazek či článek) je částí rodiče (svazku). Jakého typu jsou podřízené záznamy daného uzlu je určeno typem *titulu*, musí být však možno implementovat možnost volby typu podřízeného

titulu uživatelem.

Jelikož tuto aplikaci užívají dvě různé skupiny uživatelů k různým účelům, existují dvě verze této aplikace. První verzi – určenou k zadávání – užívají běžní pracovníci k pořizování rejstříků a slouží co k nejrychlejšímu pořizování velkého množství nových záznamů. Druhou verzi, určenou pro kontrolu, užívají převážně osoby kontrolující již zadaná data (tedy vedoucí projektu či jím pověřená osoba). Při přechodu z jedné verze aplikace do druhé je zachován označený prvek ve stromu.

V zadávací verzi se při volbě záznamu ve stromě se v pravé části zobrazí formulář pro vložení podřízeného záznamu. Pokud je záznam vložen, zobrazí se formulář pro vložení nového záznamu – a to je-li to možné tak záznamu podřízeného vloženému záznamu, v opačném případě záznamu podřízenému rodiči vloženého záznamu. Pokud chce uživatel již zadaný článek upravit, může ho ve stromu zvolit – v té chvíli bude zobrazen formulář zobrazující zvolený záznam, nabízející možnost záznam upravit či smazat. V této variantě modulu nelze upravovat svazky - při jejich zvolení se zobrazí formulář pro vložení podřízeného záznamu (svazku či článku).

Ve variantě aplikace určené pro kontrolu dat se při volbě záznamu ve stromě v pravé části obrazovky zobrazí formulář zobrazující tento záznam s možností jeho úpravy či smazání.

Podezřelé záznamy

- Svazek či článek se považují za podezřelé, pokud v nadřízeném svazku existuje záznam stejného názvu.
- Článek se považuje za podezřelý, pokud se jeho stranový rozsah překrývá s jiným základním článkem (ze stejného svazku) na více než jedné straně (pokud se překrývá na jedné straně, pak na té straně může jeden článek začínat a druhý končit; pokud je to strana zprostřed článku, pak musí být jeden z článků pouze na jedné straně a je to pravděpodobně např. sloupek či redakční vsuvka. Takové články se v časopisech vyskytují poměrně běžně a tedy nejsou považovány za podezřelé).

Rozsah zadávaných dat U svazků se zadává pouze název svazku a její kód. U numerických údajů se kód nemusí zadávat a bude generován (tak aby zachovával přirozené třídění) automaticky popř. na jeho generování bude k dispozici patřičný nástroj. Aplikaci je možné rozšířit o pořizování klíčových slov či dalších libovolných vlastností svazků (více viz kapitola 4.4.1 na straně 42). Stránkový rozsah dílů titulu se odvodí z článků náležících do svazku, proto není třeba zadávat jeho stranový rozsah.

U článků se zadává název, popřípadě český název, stránkový rozsah a pseudonym autora. Pseudonym autora lze vybrat z již zadaných pseudonymů či zadat pseudonym nový. k článku lze připojit i poznámku (např. o nečitelném nadpisu, či ujištění, že je podezřelý záznam v pořádku).

Modul *Autoři*

V tomto modulu tvoří záznamy stromu jednotlivé osoby, jejichž potomky jsou všechny zadané pseudonymy této osoby. Potomky pseudonymů jsou články, podepsané daným pseudonymem. Zvolením dané osoby, pseudonymu či článku se zobrazí formulář na úpravu vlastností dané osoby: pro článek stejný jako v aplikaci *Evidence svazků a článků*,

u osoby se editují její vlastnosti (viz. kapitola 4.3.4 na straně 37), pseudonymu lze navíc přiřadit či změnit asociovanou osobu. Nadřízenými uzly záznamů osob jsou jednotlivá písmena abecedy, která sdružují osoby s příjmením začínajícím stejným písmenem (pro rychlejší nalezení patřičného autora). Nové osoby či pseudonymy pomocí rozhraní tohoto modulu také přidávat.

Při zvolení článku se zároveň zobrazí první strana článku: to slouží ke kontrole, zdali daný článek je opravdu napsán danou osobu; ve formuláři článku lze popř. standardní metodou založit nový pseudonym a ten pak přiřadit jiné osobě.

Modul *Klíčová slova a anotace*

Tato aplikace má opět dva pohledy na data: prvním z ní je editace samotných klíčových slov, kdy lze klíčová slova zakládat, rušit, měnit, popř. je *hierarchizovat* – přiřazovat klíčové slovo jako potomka jiného klíčového slova (viz kapitola 7 na straně 16). Při úspěšné editaci klíčového slova se zobrazí formulář pro vytvoření nového klíčového slova. V tomto pohledu na data je strom tvořen klíčovými slovy - ke každému klíčovému slovu je do stromu pro kontrolu připojen seznam článků.

Druhým rozhraním je rozhraní pro přiřazování klíčových slov k článkům. Strom vypadá stejně jako při zpracovávání rejstříků, ve formuláři je možnost přiřadit k (popř odstranit od) článku klíčové slovo, při úspěšné editaci je zobrazen stejný formulář pro následující článek ve stromu, což umožňuje efektivní zadávání klíčových slov.

Anotace článků (přiřazování hodnot k danému článku či knize) bude implementována stejně jako přiřazování klíčových slov s tím, že k danému klíčovému slovu přiřazenému k článku půjde navíc přiřadit hodnotu. Zároveň bude možno u klíčového slova označit, zdali toto slovo může či musí nést hodnotu. Zatím nebyla vyžadována implementace anotací, proto zatím nebyla zahrnuta do ovládacího rozhraní. Pro tuto funkčnost jsou však připraveny patřičné datové struktury, potřebná logika i komponenty.

Modul *Správa uživatelů*

V tomto modulu jsou ve stromě (spíše listu) zobrazeni všichni uživatelé, u kterých lze měnit jméno, heslo, oprávnění, zakládat je a rušit je, stanovovat jim omezení na projekt a sledovat jejich celkový počet zpracovaných článků. Zobrazení je uzpůsobeno pro přidávání případných dalších statistik dle požadavků vedoucího projektu.

Modul *Další nástroje a vyhledávání*

Tento modul slouží jako zastřešující modul pro různé jednoúčelové úlohy (správu databáze apod.). Zároveň umožňuje vyhledávat mezi články svazky dle svého klíče (sloupce `id`) v databázim, popř. autory dle příjmení či klíče. Možnosti prohledávání lze na žádost uživatelů snadno rozšířit.

V tomto modulu se zobrazuje pouze pravá strana obrazovky, strom se seznamem záznamů je skryt. Tento modul je zobrazen také v okamžiku, kdy některý z ostatních modulů nemá vlastní obsah pro pravou část okna (např. protože není vybrán žádný záznam ve stromu záznamů): v tomto případě se tento modul zobrazí pouze v pravé části okna, v levé zůstane zobrazen strom s možností vybrat záznam. Pokud uživatel vybere záznam ve stromu, modul *Další nástroje a vyhledávání* se ignoruje, v opačném případě se ignoruje hostitelský modul.

4.4.2 Aplikace na zveřejnění digitalizovaných dokumentů

Aplikace pro zveřejnění, přístupná libovolnému uživateli internetu, slouží ke zveřejnění výsledku projektu *Depozitum* – tedy ke zveřejnění digitalizovaných dokumentů poté, co je dokončena jejich digitalizace a pořízeny rejstříky.

Dle analýzy v kapitole 2.5 na straně 19) jsou pro tuto aplikaci vytvořeny dvě verze ovládacího rozhraní: základní rozhraní, které bude sloužit náhodnému návštěvníkovi a poskytuje základní funkce co nejintuitivnějším způsobem, a rozšířené rozhraní, které poskytne maximum funkcí dostupných co nejrychlejším způsobem. Toto rozhraní je cílené na pravidelného uživatele systému.

Základní ovládací rozhraní

Základní ovládací rozhraní má svoji filosofii postavenou na volbách: v každém kroku jsou uživateli předkládány možnosti – jejich výběrem se uživatel dostane k požadovanému dokumentu. Tyto volby jsou uspořádány ve stromové struktuře, uživatel má tedy vždy možnost vybrat některou z větví stromu daného uzlu, nebo se vrátit libovolný počet úrovní zpět k vrcholu stromu. Dále jsou vždy přístupné volby z vrcholu stromu, reprezentující různé pohledy na data. Schéma možných voleb je zachyceno na schématu 4.3 na straně 44.

V kořeni ovládacího stromu má uživatel na výběr, zdali chce prohlížet digitalizované dokumenty dle jednotlivých titulů, autorů, nebo klíčových slov. Další možností, přístupnou z libovolné úrovně stromu, je možnost vyhledávání článku, a nápověda k uživatelskému rozhraní.

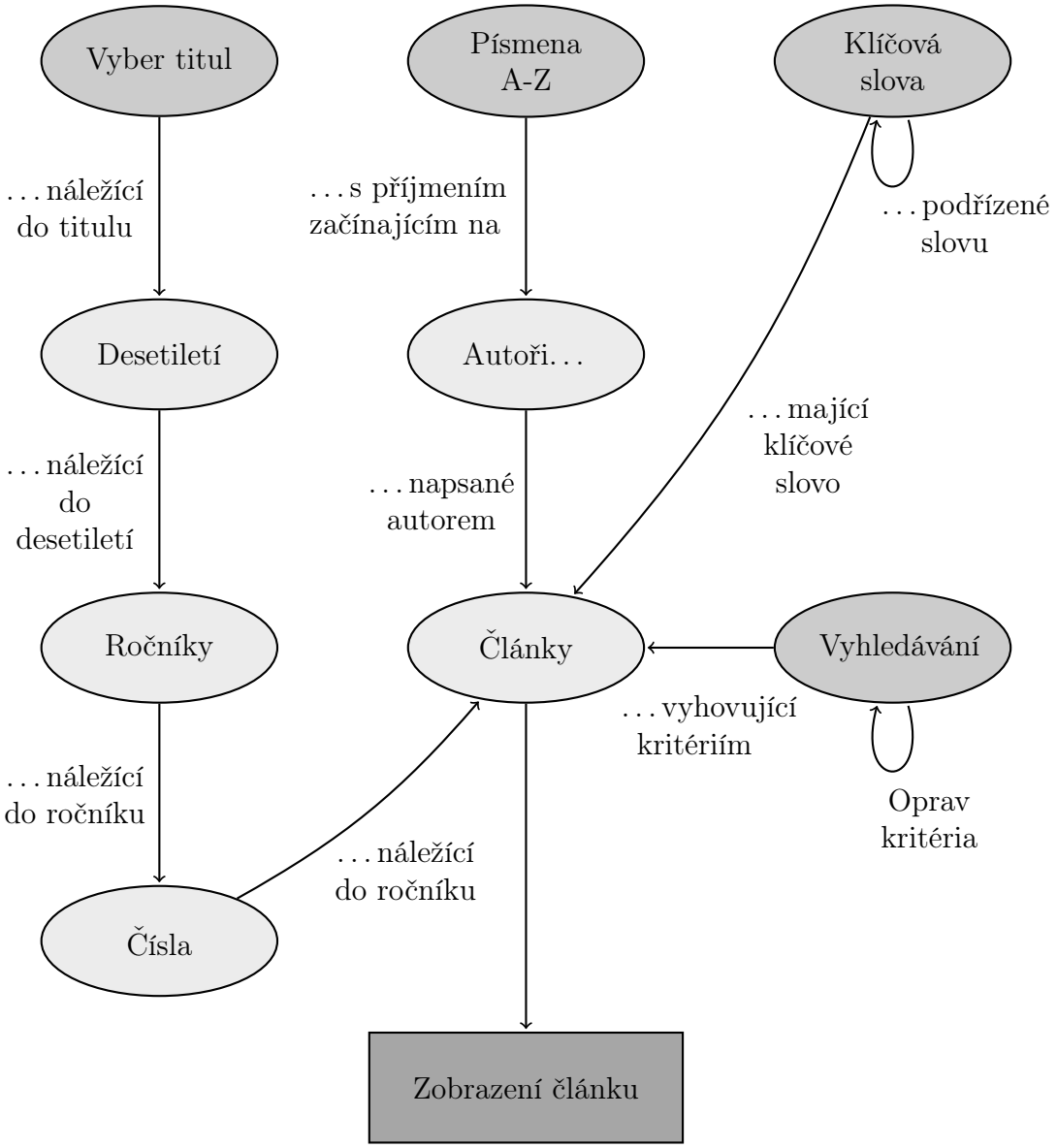
Nezávisle na tom, dle kterého kritéria uživatel záznamy prohlíží, vždy při volbě záznamu typu článek se uživateli otevře nové okno, ve kterém jsou uvedeny informace o daném článku a za nimi jsou zobrazeny všechny stránky daného článku. Toto okno je zároveň připraveno k snadnému tisku na tiskárně.

Prohlížení dle titulů V tomto pohledu na data se v základní úrovni zobrazí všechny tituly v databázi. Aplikace však musí být připravena na vnoření jedné či více úrovní nad výpis samotných titulů, omezující množství vypsaných titulů (např. písmena A–Ž, v podřízené úrovni se vypíší pouze tituly začínající daným písmenem).

Při zvolení svazku získá uživatel možnost zvolit libovolný z článků náležících přímo do daného svazku, a svazky, které jsou přímou částí daného svazku.

Prohlížení dle autorů Při zvolení této možnosti jsou uživateli nabídnuty písmena A až Z – při volbě libovolného z nich se zobrazí seznam autorů s příjmením začínajícím od tohoto písmene. Při volbě autora se zobrazí seznam všech článků daného autora.

Prohlížení dle klíčových slov Po volbě této možnosti má uživatel na výběr ze seznamu klíčových slov, které nejsou zahrnuty jako specializace (viz kapitola 7 na straně 16) jiného klíčového slova. Při volbě libovolného klíčového slova pak jsou uživateli dány na výběr všechny články mající přiřazené dané klíčové slovo a všechna klíčová slova danému podřízená danému klíčovému slovu.



- Každá hrana reprezentuje možnost zvolit jeden z mnoha záznamů daného typu. Rozvinutím hran získáme strom, zobrazený v rozhraní pro pokročilé.
- Pro kompletní schéma základního uživatelského rozhraní by bylo třeba doplnit hrany „zpět“ ke koření stromu. V obou typech rozhraní je také v každém okamžiku možnost přejít do libovolného kořene stromu (libovolného tmavého oválu).

Obrázek 4.3: Schéma aplikace pro zveřejnění digitalizovaných dokumentů

Vyhledávání V aplikace je možnost vyhledávat dle názvu článku, názvu titulu, autora článku, klíčového slova, fulltextu či přiřazené anotace. V základním vyhledávání přímo přístupném z libovolné úrovně stromu je pouze jedno kritérium, které umožňuje vyhledávat v libovolné z uvedených možností, při vyhledávání nebo při volbě *rozšířené vyhledávání* se pak zobrazí výčet všech možností vyhledávání. Po vyhledávání se zároveň zobrazí seznam všech vyhledaných článků.

Nápověda V aplikaci je také přístupná nápověda, popisující možnosti aplikace a její ovládací rozhraní.

Rozšířené ovládací rozhraní

Rozšířené ovládací rozhraní má stejný layout webové stránky jako aplikace pro pořizování rejstříků (viz schéma 4.1 na straně 39). Ve *stromu* jsou zobrazeny (podle volby patřičné položky v menu), v pravé části buďto seznam podřízených členů, nebo další informace o daném uzlu. Z menu je přístup k vyhledávání (stejného jako u jednoduchého ovládacího rozhraní), po vyhledávání se místo stromu zobrazí seznam vyhledaných článků. Dále je v menu možno zvolit zobrazení nápovědy k ovládacímu rozhraní aplikace.

Dle volby uživatele v menu může strom obsahovat následující položky, odpovídající schématu 4.3 na straně 44). Strom rozšířeného ovládacího rozhraní tedy odpovídá stromu voleb základního rozhraní:

> **Svazky** V kořeni stormu jsou všechny tituly. Potomky titulů jsou podřízené svazky, u *čísel* pak jednotlivé články. Pokud dojde k nárůstu počtu titulů, lze nad tituly umístit další úroveň stromu, třídící tituly dle vhodného (dle skladby titulů) kritéria, např. začáteční písmeno titulu, typ titulu, stáří titulu apod...
>
> **Autoři** V kořeni stromu jsou písmena A – Z, jejich potomky jsou autoři jejichž příjmení začíná daným písmenem, potomky autorů jsou články daného autora.
>
> **Klíčová slova** Ve stromu jsou klíčová slova dle svoji hierarchie, u každého klíčového slova jsou mimo jeho podřízených klíčových slov navíc uvedeny všechny články s přiřazeným klíčovým slovem.

Zobrazení článku Je-li ve stromu vybráno zobrazení článku, zobrazí se v pravé části okna seznam stránek, při volbě stránky se zobrazí patřičná stránka – mezi těmito stránkami lze tedy volně listovat, zároveň je zde možnost i přeskočit na další a předchozí článek dle pořadí v časopise, náleží-li článek do *seriálu*, tak i na další a předchozí díl *seriálu*.

Zobrazeným digitalizovaným stránkám lze měnit jejich velikost. Je možnost zobrazit souhrnné informace o článku včetně informací, kterak z daného článku citovat a přímý odkaz na daný článek. Zároveň je zde uživateli dána snadná možnost vytisknout kompletní článek.

Kapitola 5

Řešení

V této kapitole si popíšeme konkrétní realizaci hardwarového a softwarového řešení projektu *Depozitum*. Popisovaný systém na digitalizaci dat je v současné době pořád ve vývoji – jelikož některé práce pokračovali rychleji, a některé se naopak ukázali obtížnější, než jaký byl prvotní odhad cíle a rozsah projektu se během času rozšiřovali (původní účel byla digitalizace jediného časopisu), a naopak některé fáze projektu se díky své (časové i finanční) náročnosti byly odloženy, např. převádění digitalizovaných dat do textové podoby apod.

Proto jsou i jednotlivé části systému v různé části implementace: zatímco modul aplikace pro zadávání sloužící k zadávání rejstříků je již vyzkoušen a upraven dle zpětné vazby uživatelů, některé části (např. modul pro přiřazování pseudonymů k autorům) jsou právě nasazovány do reálného provozu, modul na přiřazování textu jednotlivým článkům je pouze ve fázi návrhu, neboť zadavatel tuto část projektu odložil a napsání tohoto modulu nepožadoval.

5.1 Hardwarové a softwarové řešení

Pro zpracovávaná data bylo třeba rozhodnout o vhodném software a hardware, na kterém poběží jednotlivé části systému *Depositum*.

5.1.1 Základní softwarové a hardwarové nástroje

Jako programovací jazyk pro potřebné nástroje a aplikace byl zvolen jazyk *PHP 5.0*[1]. Tento jazyk byl zvolen hlavně díky požadavku rychlých prvních výsledků, které by šly užít k prezentaci projektu a získávání dalších zdrojů pro jeho rozvoj, zároveň je tento jazyk na Katolické teologické fakultě užíván k jiným projektům a tak byla fakulta pro tento jazyk schopna (ve velmi krátké době) poskytnout nástroje pro vývoj (webový server včetně nainstalování potřebných doplňků apod.) a zároveň měla fakulta možnost pokračovat ve vývoji při případném přerušení naší spolupráce. Z těchto důvodů byl tedy zvolen jazyk *PHP*, i když takto rozsáhlému projektu by více vyhovovaly jiné jazyky, např. jazyk *Java*[2].

[1] `http://www.php.cz`

[2] `http://java.sun.com/docs/books/jls/download/langspec-3.0.pdf`

Pro webovou aplikaci sloužící k pořizování rejstříků slouží dedikovaný server s operačním systémem *GNU/Linux*[3] a webovým serverem *Apache*[4]. Na vyhrazeném serveru běží tyto tři verze aplikací pro zadávání i zveřejnění.

Testovací verze slouží k testování nových verzí systému. Používá vlastní instanci databázového serveru i úložiště digitalizovaných dat. Je přístupná pouze pro přihlášené uživatele.

Interní verze slouží k pořizování rejstříků a další správu systému a ke kontrole dat před zveřejněním. Je přístupná pouze pro přihlášeného uživatele. V této verzi jsou přístupné i rozpracované a nezveřejněné digitalizované dokumenty.

Veřejná verze Tato verze má veřejně přístupné rozhraní pro prohlížení digitalizovaných dat, rozhraní pro pořizování indexů je zcela skryto. Sdílí s interní verzí úložiště digitalizovaných dat a databázový server, publikuje však pouze zvolené dokumenty (či v případě potřeby prezentace dílčích výsledků jejich část).

Pro lepší zabezpečení a ochranu dat by bylo lepším řešením, kdyby veřejná verze měla vlastní databázi a tak by při potenciálním průniku do systému nebyla ohrožena primární data. Jelikož však v pořizovaných rejstřících (a popř. textech rozpoznaných pomocí OCR) se vyskytují chyby, jejichž eliminace by byla poměrně náročná (úplná kontrola rejstříku by zabrala cca stejný čas jako jeho pořízení), zvolila se tato varianta, která umožňuje rychlé opravy chyb dle upozornění uživatelů. Proto jsou data z databáze jsou jednou za den zálohována.

Datová úložiště pro obrazová data

Pro uložení samotných digitalizovaných dat je užito standardní datové úložiště v síti Katolické teologické fakulty – síťové disky na interních serverech této fakulty.

Primární data jsou uloženy způsobem, jež je (formátem dat, konvencí, umístěním úložiště…) vhodný pro dlouhodobé zpracování, naopak pro aplikaci zpřístupňující digitalizované dokumenty je primárním kritériem snadný a co nejrychlejší přístup k souborům. Uložiště primárních kopií by mělo být co nejvíce zabezpečeno a tedy by k němu neměl být (alespoň přímý) přístup z internetu.

Z těchto důvodů je tedy vhodné, aby aplikace pro zveřejnění digitalizovaných dokumentů nepřistupovala přímo k primárnímu úložišti, nýbrž si udržovala vlastní kopie obrazových dat.

Jelikož data poskytovaná na internetu mají daleko nižší kvalitu a tedy i velikost, toto „dublování“ kopií zvětší nároky na diskový prostor pouze minimálně. Navíc po dokončení digitalizace daného titulu lze zmenšené originály v primárním úložišti odstranit, neboť jdou kdykoli vygenerovat znovu.

Datové úložiště „veřejných kopií“ Pro uložení těchto kopií byly testovány dvě varianty: uložení v v databázi a uložení ve filesystému. Pro uložení do databáze mluví především tyto důvody:

[3] `http://www.gnu.org/`
[4] `http://httpd.apache.org/`

- Snadný přístup a manipulace se „soubory“, vytváření kopií souborů apod. Ve filesystému je třeba soubory z důvodu rychlosti členit do různých podadresářů. To s sebou nese nutnost tyto adresáře vytvářet a mazat.
- Kompaktnost uložených dat (společné úložiště s rejstříky), možnost referenční integrity, jednodušší správa obrazových dat.
- Snadná implementace generování náhledů na obrázky v reálném čase: z důvodu bezpečnosti není vhodné, aby webový server měl oprávnění zapisovat do filesystému, přístup do databáze s vhodně nastavenými právy je považován za bezpečnější.

Přes výše uvedené výhody uložení obraozvých dat do databáze bylo nakonec po testech zvoleno uložení do filesystému. Jedním z důvodů pro uložení obrazových dat do filesystému je rychlost přístupu k souborům. Tato rychlost není až tak dána rychlostí *MySQL*, ta dle testů pravděpodobně díky rychlému vyhledávání a udržování dat tabulek v paměti není pomalejší než filesystém – problémem se ukázalo nízká výkonnost *PHP* při přístupu do databáze. Toto úzké hrdlo by šlo obejít vybudováním speciálního *CGI skriptu*[5] či modulu pro webový server, poskytujícímu data – zde by však se již ztratila jednoduchost přístupu do databáze.

Další výhodou uchování dat ve filesystému je oddělení obrazových dat (jejichž originály jsou bezpečně zálohované) od rejstříků – vzhledem k rozdílné velikosti rejstříků a samotných obrazových dat se tak některé manipulace se systémem zjednodušují, je snadná i implementace několika verzí rejstříků, sdílející jedny obrazová data.

Název souboru strany v aplikaci Aby bylo získávání obrazových dat co nejrychlejší, bylo možno za běhu aplikace měnit *kód* jednotlivým svazkům apod. nejsou soubory stran v datovém úložišti webového serveru uložené pod svým kódovým jménem, nýbrž pomocí celého čísla – `id` strany – užívaného v databázi jako primární klíč v tabulce stran (viz kapitola 5.1 na straně 52).

V tomto úložišti je stanoven adresář, ve kterém jsou uloženy všechny soubory stran. V tomto adresáři jsou adresáře `original` a `thumbnail` obsahující plnou velikost obrazu (tzn. obrázek o délce hrany 1200px), respektive zmenšeninu o delší hraně dlouhé 100px sloužící k zobrazení náhledu strany.

Pro běžné filesystémy není příliš vhodné užívat strukturu souborů s velkým počtem položek v jednom adresáři, neboť při vyhledávání souboru užívají lineární vyhledávání (např. filesystém *FAT32*[6] či *Ext2*[7]/*Ext3*[8]). Ostatní filesystémy používají většinou kombinace stromů a hashování s řetězci, které při nedostatečné délce hashovacího klíče (ten je např. v *XFS* [24] čtyři byty) také využívá lineárního prohledávání, proto je vhodné limitovat počet souborů v adresáři. Proto je pro uložení souboru strany užito následující schéma:

Je-li `id` id strany, pak je daná strana uložena pod názvem `X.jpg` v podadresáři `Y` adresáře `original` respektive `thumbnail`, kde

$$\mathtt{Y} = \frac{\mathtt{id}}{c} \qquad \mathtt{X} = \mathtt{id} - c\,\mathtt{Y} \tag{5.1}$$

[5] `http://www.w3.org/CGI/`

[6] http://download.microsoft.com/download/1/6/1/161ba512-40e2-4cc9-843a-923143f3456c/fatgen103.doc

[7] http://e2fsprogs.sourceforge.net/ext2intro.html

[8] http://www-128.ibm.com/developerworks/linux/library/l-fs7.html

Úložiště souborů stran by mělo pokud možno užívat filesystém, který neprohledává obsah adresářů lineárně, možné užít je např. souborový systém *XFS*, nebo *EXT3* s rozšířením *dir_index* [25]; někdy bývá pro tyto účely doporučen i *ResierFS* [16, 7, 26]. Pokud bude užito filesystému s lineárním prohledáváním adresářů, (např. *FAT32*) by mělo být uvažováno o snížení konstanty c a vybudování další úrovně (či dalších úrovní) hierarchie podadresářů.

5.1.2 Datové úložiště pro rejstříky

Pro uložení rejstříků byla jako datové úložiště zvolena databáze. Z dostupných SŘDB s vhodnou licencí byl zvolen SŘBD *MySQL*[9] s jádrem pro ukládání dat (engine) *InnoDb*[10], poskytujícím transakce a referenční integritu. Nicméně pro objemy dat, které se dosud i v dohledné době pořizují by dostačovala pravděpodobně libovolná z z dostupných databází s volnou licencí (dále byly zvažovány *Postrges*[11] či *Firebird* [12]).

Vhodnost *MySQL* pro větší projekty O vhodnosti či nevhodnosti toho kterého SŘBD jsou mezi odbornou veřejností spory bez jednoznačného výsledku. Jako nevýhoda *MySQL* bývá uváděno nižší podpora standardů jazyka *SQL*[13] a někdy i horší škálování výkonu při přístupu mnoha uživatelů. Poslední vytýkanou vadou *MySQL* je údajná nestabilita a špatná crash-recovery. Tato informace není podložená žádným důvěryhodným zdrojem, pouze se cituje na různých diskuzních fórech (např. [50]).

Naopak výhodou *MySQL* je snadná implementace replikace databáze a výkon databáze při čtení. Jelikož verze *MySQL 5.0* většinu velmi pokročila [5] v implementaci standardů[14], po pořízení databáze (při kterém zpravidla nepracují dva lidé nad jedněmi daty) se bude z databáze pouze číst a při růstu projektu bude pravděpodobně potřeba nějaká forma replikace či rozložení zátěže, zdá se být *MySQL* dobrou volbou.

API pro přístup do databáze Aby bylo v případě potřeby možné nahradit *MySQL* jiným *SŘBD*[15], je v *PHP* vystavěna nad *MySQL* abstraktní databázová vrstva, která při změně databáze redukuje potřebné změny na minimum. Implementace této databázové vrstvy pak používá knihovnu *mysqli*.[16]

Jednotlivé knihovny Jelikož digitalizace každého titulů je většinou hrazena pomocí jiných zdrojů (různé granty, externí spolupráce apod.), je digitalizace každého titulu v jiné fázi. Proto je projekt v současné době rozdělen na jednotlivé *knihovny* – v každé knihovně je zpravidla jeden *titul*. Každá knihovna má přiřazenou svoji vlastní databázi.

Výhoda tohoto uspořádání je snažší záloha jednotlivých titulů, snadné zveřejnění již hotových titulů a snažší správa celého projektu. Zároveň je systém postaven tak, že každé

[9] `http://www.mysql.org`

[10] `http://www.innodb.com/`

[11] `http://www.postgresql.org/`

[12] `http://www.firebirdsql.org/`

[13] `http://en.wikipedia.org/wiki/SQL`

[14] Bohužel nikoli všechny, např. stále nelze provést v příkazu *UPDATE* vnořený *SELECT* do měněné tabulky

[15] `http://http://cs.wikipedia.org/wiki/Syst%C3%A9m_%C5%99%C3%ADzen%C3%AD_b%C3%A1ze_dat`

[16] `http://cz.php.net/mysqli`

knihovně lze přiřadit jiná verze (kódu) aplikace – pokud je započata digitalizace nového titulu, který vyžaduje v aplikaci změny, je možné mu vyhradit speciální verzi aplikace, a až po důkladném otestování sloučit tyto dvě verze aplikace zpět do jedné. Další výhodou je snadná obnova dat: pokud je jedna databáze porušena, při menším počtu záznamů v databázi a uživatelů do databáze zapisujících je snadné najít ještě neporušenou verzi zálohy, popř. z porušené verze extrahovat změny, které nevedly k porušení databáze a do databáze obnovené zálohy je přidat.

Nevýhodou tohoto uspořádání je separace jednotlivých titulů a tedy možnost vyhledávat pouze v rámci jedné knihovny. Zatím jsou digitalizované tituly natolik odlišné povahy, že tato separace není příliš na škodu (tato separace byla také přáním zadavatele). V případě potřeby však není problém jednotlivé knihovny sloučit.

Sloučení knihoven může být jak faktické (přesun všech dokumentů do jedné databáze), nebo lze jednotlivé knihovny sloučit vytvořením zastřešujícího vyhledávače. V druhém případě by bylo velmi vhodné, aby tento vyhledávač pracoval s některým ze standardních formátů (viz kapitola 2.1 na straně 8), neboť by to umožnilo integraci dalších digitalizovaných dokumentů poskytovaných dalšími subjekty.

5.1.3 Zálohování a obnova dat

Uložiště primárních dat je (kromě standardní zálohy pomocí technologie *RAID 5*[17]) zálohováno standardními postupy zálohování dat na Katolické teologické fakultě. Mimo to vždy po dokončení digitalizace titulu je (digitální) kopie tohoto titulu přehrána na externí pevný disk. Ten je pak (fyzicky) uložen na jiném místě než samotné servery, což snižuje možnost ztráty dat při požáru či jiné živelné katastrofě.[18] Přístup do datového úložiště, stejně jako k vytvářeným zálohám, má pouze vedoucí projektu. Při scanování se digitální kopie ukládají na disky pracovních stanic na Katolické teologické fakultě (ty jsou zálohované spolu s celou sítí této fakulty) a v dohodnutém intervalu s vedoucím práce mu odevzdávají výsledky.

Pokud by byl tento projekt zastaven či ukončen, vzhledem k překotnému vývoji v oblasti informatiky hrozí, že by pořízená data mohla být za několik (desítek) let nečitelná (neboť by již nebyla ve formátu v té době podporovaném). Aby se toto riziko zmenšilo, je na pevné disky po nakopírování záloh nainstalován operační systém a prohlížeč zálohovaných dat. Při případném zakonzervování projektu by k těmto zálohám měl být připojen i počítač, který by byl schopen s těmito disky komunikovat.

Webový server, který obsahuje kopie obrazových dat pro zveřejnění, je taktéž zálohován pomocí interních fakultních mechanismů. Jelikož kopie digitálních dat se dají snadno znovu vygenerovat, nejsou již dále zálohovány.

Zálohy rejstříků uložených v databázi jsou nejcennější data získané během digitalizace. Navíc u nich hrozí, že budou např. chybou uživatele porušena; tato chyba se může odhalit až za delší dobu. Proto je třeba kromě každodenních záloh prováděných v rámci zálohování fakultních serverů uchovávat i starší zálohy: jednou denně (pomocí plánovače úloh `cron`) je zazálohován obsah všech databází obsahujících produkční data. Tyto zálohy jsou rotovány, přičemž jsou zachovávány zálohy za posledních čtrnáct dní a záloha z prvního dne každého měsíce. V současném rozsahu dat je pro zálohu obrazových dat

[17] `http://en.wikipedia.org/wiki/Standard_RAID_levels`

[18] Zatím jsou však disky uloženy v druhém křídle jedné budovy, což je nedostatečné prozatimní řešení. Na hledání vhodného a finančně dostupného řešení se pracuje.

naprosto dostatečný *SQL dump*[19], při větším růstu dat by bylo třeba nasadit kompresi záloh, popř. inkrementální zálohu dat.

Data z databáze jsou navíc po dokončení digitalizace titulu (popř. při přerušení digitalizace či odložení dalších jejích kroků) přidána ve formě SQL výpisu k primárním obrazovým datům, a to jak v primárním datovém úložišti, tak i na externích discích.

Při porušení databáze rejstříků je třeba analyzovat vzniklé porušení a volit postup obnovy dat individuálně: v některém případě může být nejvhodnější postup data obnovit z poslední neporušené zálohy, v některém případě může být vhodné z porušené databáze do této obnovené vyextrahovat validní změny, případě může být nejvhodnější řešení vadné záznamy v databázi opravit či smazat. Ve vývoji je systém na přesný záznam změn vykonaných jednotlivými uživateli, který dále zjednoduší proces obnovy databáze.

Vzhledem k ceně pořizovaných dat je dosud prováděné zálohování spíše nedostatečné. Při dalším rozvoji projektu je nutné zajistit uložení přinejmenším jedné kopie záloh na geograficky odlišné místo a zvážit možnost ukládání na média, užívající jiný způsob záznamu dat než je magnetický záznam.

K bezpečnému uchování digitálních kopií pro další generace by bylo třeba vybudovat chráněné úložiště dat, které by bylo schopno přetrvat i větší živelné katastrofy či válečný konflikt apod. Na vybudování tohoto úložiště by bylo vhodné spolupracovat s dalšími subjekty zabývajícími se digitalizací. Takovýto projekt je však zatím zcela mimo finanční možnosti Katolické teologické fakulty.

Přesný popis postupu zálohování a postupu při obnově dat lze nalézt v uživatelské příručce aplikace pro zadávání dat.

5.2 Schéma databáze rejstříků

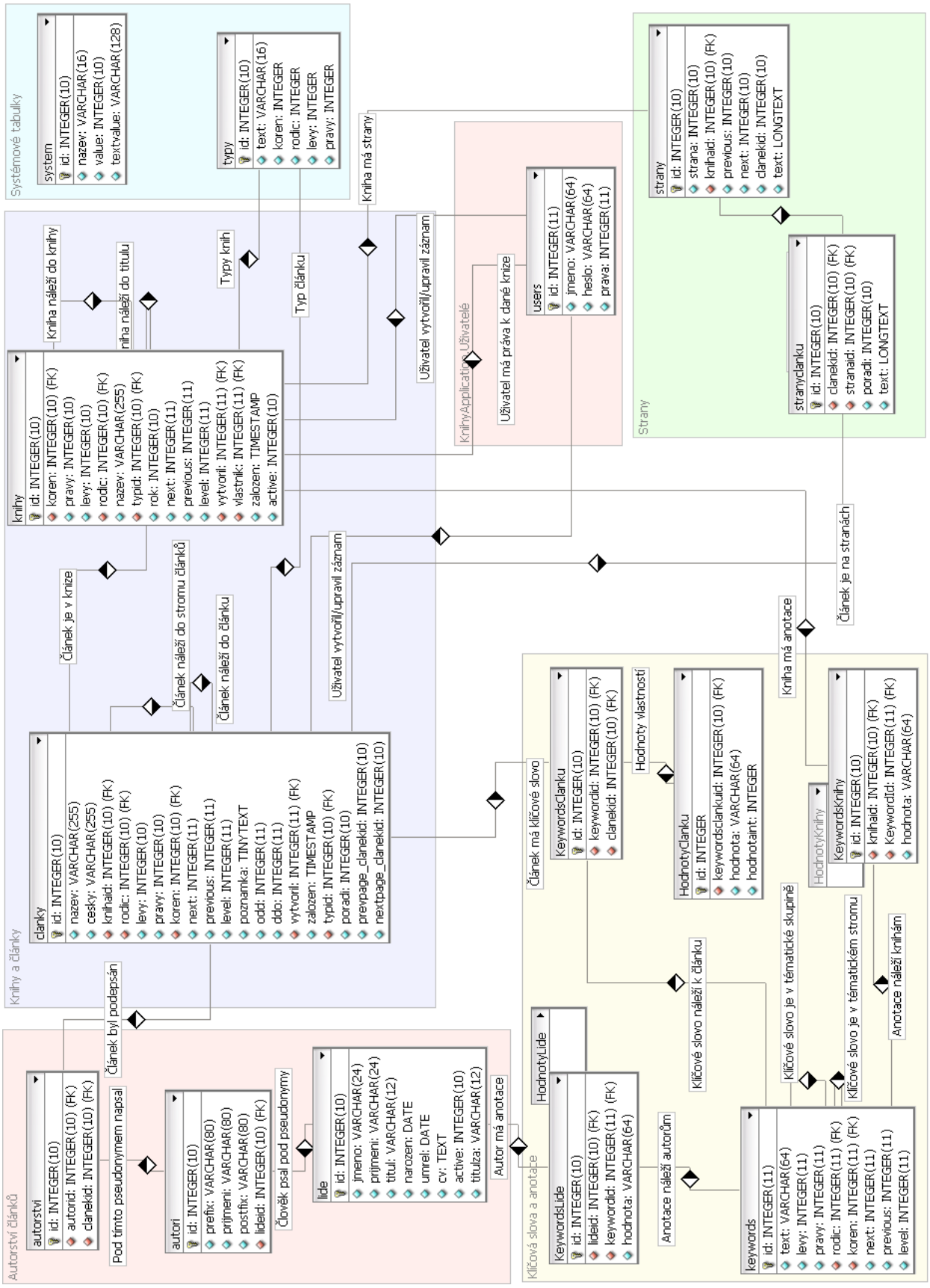
Schéma databáze užívané k uložení rejstříků a dalších metadat o digitalizovaných dokumentech je na obrázku 5.1 na straně 52. Plná dokumentace k databázi je k dispozici v programátorské dokumentaci k projektu, zde se budeme věnovat pouze základní filosofii uložení do databáze a některým zajímavým momentům.

Paradigmata

V databázi byla zvolena následující filosofie pojmenování: tabulky mají jméno v množném čísle, každá tabulka má primární klíč na sloupci s názvem *id* typu *autoinkrement*. Cizí klíče (není-li v dané tabulce více cizích klíčů do dané tabulky) mají formát `<tabulka>id`, kde `<tabulka>` je název tabulky s cizím klíčem v jednotném čísle.

Stromy V některých databázových tabulkách jsou uložená data v podobě stromu. Ty jsou do „ploché“ tabulky kódovány pomocí levopravého kódování, často označovaného jako *Nested set* či *Modified Preorder Tree Traversal Algoritmus* [44]. Tabulky, ve kterých je uložen strom, mají následující pole:

`levy, pravy`	Kód levopravého kódování.
`rodic`	Odkaz na přímého předka.
`koren`	Odkaz na vrchol stromu. Např. u tabulky *knihy* odkazuje na *titul*.
`level`	Úroveň uzlu ve stromu.



Obrázek 5.1: Schéma databáze

Jelikož pro základní funkčnost stromu je potřeba pouze odkaz na předka, a levopravé kódování vyžaduje při přidání prvku do stromu úpravu kódu v celém stromu, ostatní kódy se po zadání dat vygenerují pomocí skriptu. Dále budeme potřebovat tento termín:

Definice 11 Les *je množina stromů.*

5.2.1 Způsob uložení jednotlivých entit

Na schématu 5.1 na straně 52 jsou vyznačeny oblasti, které odpovídají jednotlivým entitám v databázi.

svazky Svazky jsou uloženy v tabulce `knihy` do lesa, kořenem každého stromu je záznam daného titulu. Relace podřízenosti ve stromu odpovídá vztahu býti části. Např. potomci časopisu jsou jeho ročníky, potomci ročníku jsou čísla a přílohy; potomci slovníku jsou díly obsahující hesla od daného dílu abecedy apod. Tento systém (narozdíl od pevné tabulky ročníků, čísel apod.) zaručuje snadnou adaptabilitu na jiné typy dokumentů.

články Všechny články, popř. seriály jsou taktéž uloženy v *lese* v tabulce `clanky`. Platí pravidlo, že existují-li v jednom svazku (či v jejích potomcích) více článků k sobě logický náležících, je pro ně zřízen *seriál* s vazbou na daný svazek, To umožňuje stručnější a přehlednější výpis seznamu článků z daného svazku.

typy Každý dokument či článek má uveden svůj typ (v číselníků typů `typy`. Tyto typy jsou opět seřazeny do stromu – to umožňuje např. vyhledat všechny svazky, které jsou typu časopis nebo jeho částí apod. Typ článku udává, zdali jde o *článek* či *seriál*, případně jde-li o seriál vygenerovaný automaticky, nebo o ruční záznam.

autoři Pro uložení dat o autorech existují dvě tabulky: s tabulkou `autori`, obsahující osoby, je relací $1 : n$ spojena tabulka všech podpisů daných lidí. Pomocí tabulky `pseudonymy` jsou podpisy daných lidí přiřazeny k jednotlivým článkům.

klíčová slova a anotace U jednotlivých dokumentů se je vhodné evidovat klíčová slova i různé informace (např. u některých svazků má smysl mluvit o edici, některé časopisy mají číslované ročníky apod.). Proto není příliš vhodné tyto informace vložit do tabulky formou pevných sloupců, místo toho je vhodné užít obecnější techniku.

Systém *Depositum* má speciální tabulku názvů hodnot `keywords`. Pomocí tabulek `keywords<...>` lze jak knize, článku, tak i autorovi přiřadit dané klíčové slovo. Pomocí tabulky `hodnoty<...>` pak lze k takto přiřazenému klíčovému slovu přiřadit jednu či více hodnot. Tyto tabulka lze zároveň užít pro efektivní vyhledání patřičných entit s danou vlastnosti či hodnotou vlastnosti. Tabulka `hodnotyknih` a `hodnotyautoru` jsou pro větší přehlednost ve schématu 5.1 pouze naznačeny).

Přiřazované vlastnosti i klíčová slova (ať již s hodnotou či bez) jsou taktéž seřazeny do stromu. To umožňuje sdružovat vlastnosti do skupin a tím je vhodně řadit při výpisu vlastností, jednak hierarchizovat klíčová slova (viz kapitola 7 na straně 16).

[19] `http://dev.mysql.com/doc/refman/5.0/en/mysqldump.html`

strany Strany náleží spolu se svazky do fyzického členění dokumentu.. Jelikož však většina dalších vlastností stran (nemají název, překladatele) je od svazků odlišná, a velikost lesa svazků by příliš narostla, jsou zařazeny do zvláštní tabulky `strany`.

U strany se udává její *číslo* (viz kapitola 4.2.2 na straně 32), zakódované do sloupce `strana` jako `S*1000+M`. Tento způsob uložení sice porušuje formální pravidla pro normální formu tabulky, urychluje však vyhledávání dané strany. Vzniklé omezení na maximální počet 999 stran s neuvedeným číslem za sebou je díky pravidlu o očíslování velkého počtu neočíslovaných stran (viz kapitola 4.2.2 na straně 32) také pouze teoretickým omezením.

Dále může být v tabulce stran uložen text získaný aplikací OCR na tuto stranu. Vazba stran k článku je uložena v tabulce `stranaclanku`, vytvářející relaci $m : n$ mezi tabulkami `strany` a `clanky`.V této tabulce také může být uložen (OCR rozpoznaný) text dané strany náležící danému článku.

5.3 Aplikace

Dle specifikace (viz kapitola 4.2 na straně 30) tvoří jádro systému *Depositum* dvě aplikace, *Aplikace pro zadávání* a *Aplikace pro zveřejnění digitalizovaných dokumentů*. Obě tyto aplikace jsou postaveny na společném základě a společných knihovnách. Aplikace jsou psány v jazyce *PHP*[46], který generuje *HTML* stránky ve formátu *HTML 4.0 transitional*[41], s výjimkou jednoduchého rozhraní (u aplikace pro zveřejnění digitalizovaných dat) zároveň užívají *javascript* (*ECMAscript* [8]) a technologie na bázi *AJAXu*[20].

5.3.1 Společné knihovny

Většina z knihoven užívaných v systému *Depositum* jsou snadno využitelné v libovolných jiných projektech, přestože byly vyvinuty či velmi zdokonaleny v rámci projektu *Depositum*. Zde popíšeme pouze základní rysy těchto knihoven a jejich zajímavých momentů, podrobnější popis lze nalézt v programátorské dokumentaci k projektu *Depositum*.

Systémová knihovna Tato knihovna je (provázaný) souhrn různých nástrojů pro řízení webové session, řízení výstupu webové stránky, a dalších nástrojů, užívaných dalšími knihovnami.

Knihovna SQL Abstraktní rozhraní pro provádění SQL dotazů. Díky této knihovně je celý systém *Depositum* nezávislý na použité databázi k ukládání rejstříků a dalších potřebných údajů.

Knihovna Widget Tato knihovna poskytuje ovládací prvky – widgety, ze kterých je poskládáno GUI aplikací (mimo jednoduchého uživatelského rozhraní aplikace).

Knihovna iterátorů Tato knihovna je (narozdíl od předešlých) vázaná na systém *Depositum*. Poskytuje objekty – *iterátory* – sloužící k enumeraci jednotivých entit v systému, např. k výčtu článků v publikovaných v daném čísle apod. Tyto iterátory jsou pak užívány objekty z ostatních knihoven k zobrazení dat.

Z uvedených knihoven se budeme věnovat pouze knihovně *Widget*, která je z uveřejněných knihoven nejkomplexnější. Zbylé jsou popsány v programátorské dokumentaci.

[20] `http://en.wikipedia.org/wiki/AJAX`

Knihovna Widget

Tato knihovna poskytuje objekty – *widgety* – ze kterých se skládá uživatelské rozhraní aplikace. Mezi tyto objekty patří (mimo jiné) *list*, zobrazující (i vícesloupcový) seznam prvků či tabulku, *strom*, zobrazující data uspořádaná ve stromové struktuře, *sliderbar*, umožňující výběr aktuálně zobrazeného prvku ze seznamu (např. stránky) či *panel*, zobrazující definovaná uživatelem widgetu.

Užité technologie Tato knihovna je postavená na technologii velmi příbuzné technologii *AHAH*[21][9], která je sama velmi příbuzná s nyní velmi známou technologii *AJAX*[22]. Rozdíl technologie *AHAH* oproti *AJAXu* spočívá v tom, že *AHAH* nezískává ze serveru místo *XML*[42] dat, podle kterých by vytvářela v klientu HTML kód, nýbrž hotové kusy *HTML* kódu. Knihovna *Widget* pak užívá modifikaci technologie *AHAH*, kdy je komunikace se serverem realizována pomocí *HTML* elementu `IFRAME` (místo `XMLHttpRequest`).

Srovnáme-li tedy uvedené technologie, lze u použité technologie vypozorovat následující výhody a nevýhody:

Srovnání s AJAXem

– Některé činnosti vyžadují součinnost serveru (např. seřazení tabulky dat lze teoreticky dělat v klientovi, naše aplikace, mající filosofii společnou s *AHAHem*, bude tuto činnost provádět zpravidla na serveru), z toho v některých případech vyplývající nižší rychlost aplikace.

– *AJAX* lze velmi snadno provazovat s dalšími webovými službami, desktopovými aplikacemi apod.

\+ V některých případech větší rychlost aplikace, neboť dnešní prohlížeče lépe zvládají manipulaci s celými kusy *HTML* než s *DOM* stromem [19].

\+ V *PHP* či *MySQL* lze některé činnosti provádět efektivněji než v *javascriptu*, v některých případech je na serveru k dispozici data, umožňující rychlejší vykonání úkolu (např. indexy pro řazení dat). Proto může být aplikace založená na *AHAHu* rychlejší, nabízet více možností či zatěžovat méně síť (neboť nemusí přenášet všechna data, nýbrž pouze výsledky).

\+ Rychlejší a přirozenější vývoj nových ovládacích prvků (*HTML* umí většina vývojářů a jsou schopni v něm rychle napsat požadovaný prvek, zatímco manipulace se stránkou pomocí *DOM* je často zdlouhavá).

\+ Větší kompatibilita se staršími browsery, neboť se nespoléhá na složité javascriptové operace.

Srovnání užití `IFRAME` a `XMLHttpRequest`

– Horší možnosti sledovat stav požadavku na server.

\+ Větší kompatibilita se staršími browsery.

\+ Jednodušší vykonávání skriptů zaslaných serverem (zatímco `IFRAME` tyto skripty provede automaticky, v datech získaných pomocí `XMLHTTPRequest` je nutno vyhledávat elementy `SCRIPT` [45]).

[21] `http://en.wikipedia.org/wiki/AHAH`

[22] `http://cs.wikipedia.org/wiki/Asynchronous_JavaScript_and_XML`

+ Jednoduché posílání více funkcí na serveru v rámci jednoho požadavku.

Užití v prohlížeči bez javascriptu Aby bylo možné knihovnu *Widget* užít i v jiných projektech, poskytuje tato knihovna i možnost běhu v prohlížečích nepodporujících javascript (a tedy nemožnost užít *AHAH*). Vzhledem k existenci jednoduchého uživatelského rozhraní, které je pro užití v prohlížeči bez javascriptu vhodnější, však tato možnost není v projektu *Depositum* použita.

Akce Každý *widget* má definované akce, které je schopen provádět, a obslužné rutiny na tyto akce. Akcí existují tři typy: akce na klientovi, akce na serveru a HTTP akce.

Akce na klientovi jsou akce prováděné v prohlížeči pomocí javascriptu, vyvolané jsou buďto voláním patřičného kódu, nebo *HTML DOM* události (např. *onclick*). Akce na serveru jsou prováděné pomocí *AHAHu*. Většinou se většinou volají z obslužných rutin akcí na klientovi, pokud není na klientovi dostatek informací k dokončení této akce.

HTTP akce jsou implementovány pro prohlížeče bez podpory javascriptu. Tyto akce způsobí obnovení strany (pomocí spuštění klasického *HTML* odkazu), před jejím samotným vykreslením však nejprve provedou danou akci.

Příkladem takové akce může být např. akce *expand* widgetu strom (typ *TTreeView* – widget sloužící k zobrazení informací ve formě stromu), která slouží k rozvinutí větve stromu. Tato akce je zavolána při kliknutí na ikonu plus u daného uzlu stromu. Ten je realizován jako odkaz, mající vlastnost *onclick*, která při kliknutí zavolá akci *expand* na klientovi. Obslužná rutina této akce se podívá, zdali daná větev již nebyla rozvinutá. Pokud ano, rozvine ji znovu. Pokud ne, zavolá akci *expand* na serveru, která vygeneruje větev daného stromu a příkaz k nahrazení obsahu dané větve vygenerovaným obsahem. Tento příkaz (spolu s vygenerovanou větví) je zaslán v odpovědi na provedenou akci zpátky klientovi, kde je vykonán.

Ať je větev na klientovi rozvinuta hned, nebo je poslán požadavek na sserver, v obou případech obslužná rutina vrátí hodnotu *false*, což způsobí neprovedení daného odkazu. okud klient nepodporuje javascript, událost *onclick* se neprovede, což způsobí otevření odkazu a vykonání *HTTP akce*.

Master-detail relationship a akce Hlavní síla této knihovny spočívá v implementaci *master-detail relationship* (*master-detail* pohledu na data). Jednotlivé widgety lze podřizovat jiným widgetům – při zvolení záznamu v nadřízeném prvku se změní obsah podřízeného prvku. Např. při volbě patřičného článku se zobrazí seznam stran tohoto článku, při volbě strany ze seznamu stran se zobrazí patřičná strana.

Master-detail relationship je realizován pomocí javascriptové akce *update* s parametrem *klíč*, identifikujícím zvolený záznam v *master* widgetu. Díky tomu lze snadno vytvořit nový typ *slave* widgetu, které budou plně spolupracovat s libovolným *master* widgetem.

Standardní odezva na akci *update* je nahlédnutí do cache, zdali pro daný záznam zvolený v *master* widgetu není již v cache uschována podoba *slave* widgetu. Pokud není, tak je na server odeslán požadavek na získání podoby widgetu (ve formě HTML fragmentu) odpovídají *klíči*. Aktuální podoba widgetu je pak v obou případech nahrazena získanou podobou widgetu, při získání podoby widgetu ze serveru je navíc do cache uložena dvojice *klíč* a podoba widgetu. *Slave* widget však může definovat odezvu na zavolání

akce *update* i zcela jinak, např. měnit atribut `src` ve svém vnořeném obrázku (HTML elementu `img`).

Pokud není povolen javascript, je při změně záznamu (kliknutí na patřičný prvek realizovaný jako odkaz) na server poslána akce *master* widgetu *select*. HTTP akce vždy způsobí překreslení celé webové stránky, akce obsluha *select* pak zajistí, že widgety budou zobrazovat uživatelem zvolený záznam.

Tato implementace *master-detail relationship* snižuje zatížení serveru – jednou získané informace jsou uchovávány na klientovi a i, pokud nejsou k dispozici data pro požadovanou akci, při jejich získání ze serveru se nemusí generovat celá webová stránka znovu. Navíc velmi zjednodušuje definování nových widgetů podporujících tento mechanismus: pro definování nového widgetu je třeba pouze vytvořit potomka obecného widgetu a implementovat metodu, která zobrazí obsah daného widgetu.

Session a persistence widgetů Jelikož je třeba provádět pomocí *AHAHu* na serveru akce jednotlivch widgetů, je třeba, aby tyto widgety byly persistentní tzn. uchovávat stav těchto widgetů (např. dle kterého kritéria je seřazen daný seznam). Proto se tyto widgety ukládají do session[23]. Pro zajištění této perzistence se však ukazuje, že samotný mechanismus session v *PHP* nedostačuje, neboť neumí rozlišovat mezi jednotlivými okny prohlížeče. Při otevření více oken prohlížeče je třeba vyřešit interferenci widgetů mezi sebou.

Řešení, kdy by každá instance widgetu měla přiřazen unikátní identifikátor a pomocí něho se rozlišovaly jednotlivé widgety v různých oknech není v tomto případě příliš vhodná – jelikož nelze detekovat zavření patřičného okna, musely by v ukládacím prostoru pro session data zůstávat definice všech vygenerovaných widgetů. To by při delší práci s více okny způsobovalo zahlcení mechanismu session. Tento problém by sice šel řešit občasným pročištěním této cache, pro další nevýhody tohoto řešení (např. nemožnost v tomto případu odkazovat mezi widgety pomocí globáního jména bez vlastnictví odkazu na tento widget) bylo zvoleno následující řešení:

Při požadavku na zobrazení strany je straně přiřazeno číslo (ze vzrůstající posloupnosti), které se ukládá do session. Pokud jsou v tu chvíli v session přítomna data widgetů, jsou spolu se svým identifikátorem strany uložené do databáze. Spolu s každým požadavek na vykonání akce (ať již *na serveru* nebo *HTTP akce*) je s požadavkem posíláno číslo strany, z které je požadavek posílán. Před zpracováním akce/í je toto číslo porovnáno s aktuálním číslem v session – je-li různé, aktuální session data náležící widgetům se uloží (spolu s aktuálním číslem strany v session), z databáze se načtou sesison data náležící widgetům z dané strany a akutální číslo strany v sesion se změní na přijaté číslo.

Znovuužitelnost knihovny Mnoho widgetů této knihovny využívá *iterátory*. Iterátor je objekt enumerující seznam položek daného typu. Daný widget (např. seznam či strom) je pak jako argument příjmá iterátor daného typu a zobrazuje jeho obsah v patřičné formě. Vzhled widgetů lze modifikovat pomocí kaskádových stylů, lze změnit i prefix pro css třídy jednotlivých *HTML* elementů dané instance widgetu. Některé widgety (např. strom či list) požadují od iterátorů název *css* tříd svých prvků, čímž umožňují snadné řízení jejich vzhledu.

Tuto knihovnu lze tedy snadno užít i v jiných projektech, neboť pro užití této

[23] `http://cz.php.net/manual/en/book.session.php`

knihovny stačí do projektu vložit soubor s implementací daného widgetu (příkazem *require_ once*) a implementovat několik metod patřičného iterátoru.

5.3.2 Aplikace pro zadávání

Tato aplikace používá klasické HTML formuláře (definované v [41]) pro zadávání, editaci a mazání záznamů v databázi. Jádrem aplikace je knihovna *FMForms*, sloužící k zápisu záznamů do databáze a jejich editaci.

Knihovna FMForms

Pro rychlé vkládání rejstříku je klíčové, aby pracovníci měli k dispozici kvalitní nástroj na pořizování rejstříků. Jelikož projekt *Depozitum* je v setrvalém vývoji a různé dokumenty vyžadují různý rozsah informací ukládaných do databáze, je třeba, aby bylo možno snadno modifikovat (popř. vytvářet nové) formuláře, které slouží k zadávání jednotlivých dat. Proto byla v rámci projektu vyvinuta knihovna *FMForms*.

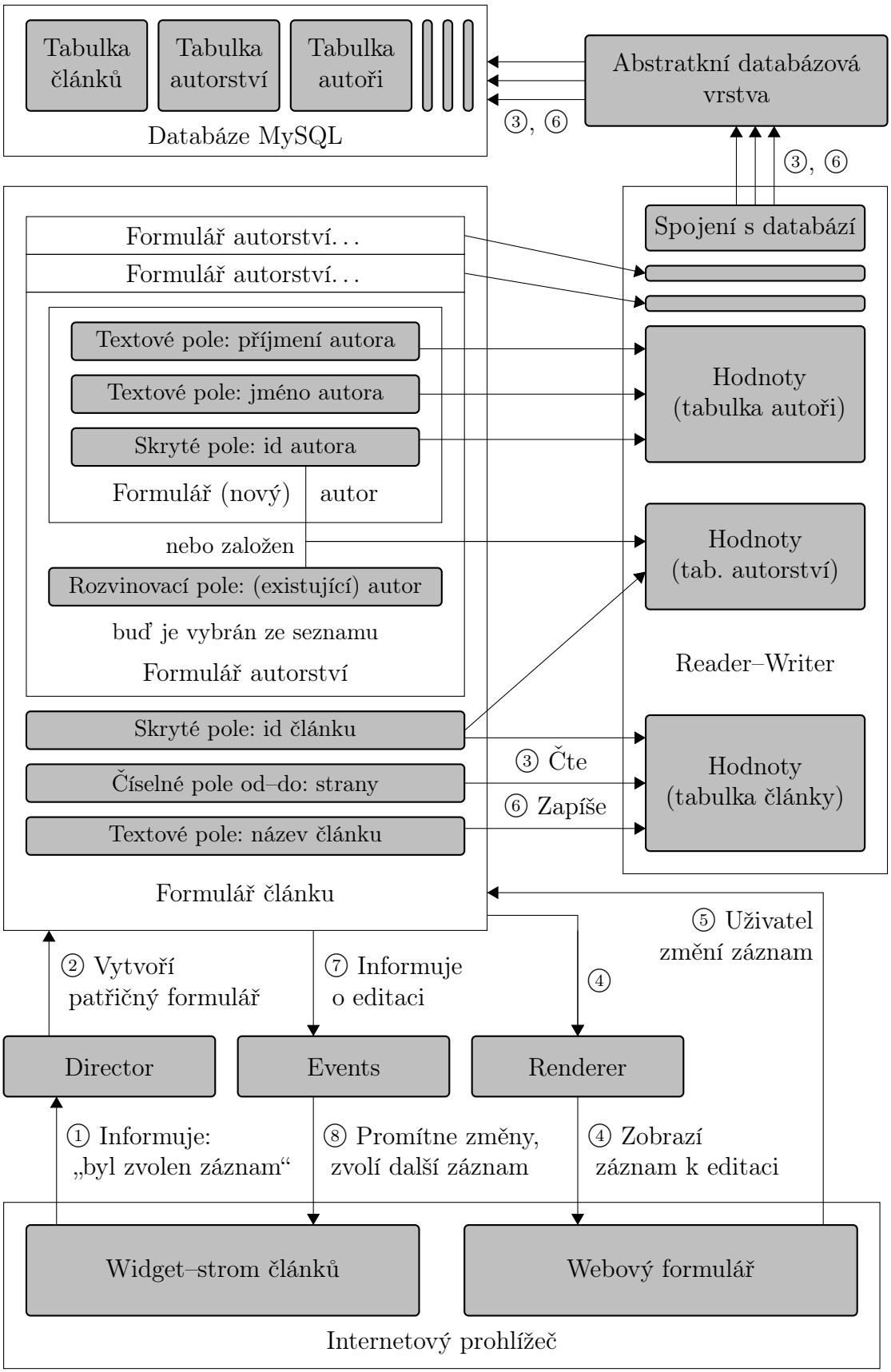
Tato knihovna slouží k definici a vytváření formulářů, sloužících k zadávání či editaci záznamů do databáze, jejich zobrazování a zápisu dat zadaných uživatelem do databáze či jiných datových zdrojů. Příklad definovaného formuláře, spolu s dalšími objekty z této knihovny (sloužícími k zápisu dat do databáze, vykreslení formuláře a interakci formuláře s okolím) a principem funkčnosti je na schématu 5.2 na straně 59.

Knihovna je v základu postavena na klasických *HTML* formulářích a je kompatibilní s libovolným prohlížečem podporujícím *HTML 4.0*. k validaci dat na straně klienta používá knihovna javascript (pokud není povolen, validace proběhne až po odeslání formuláře na server). Aby bylo načtení formuláře co nejrychlejší, lze některá data (např. nabídky rozvinovacích seznamů) načítat pomocí technologie *AHAH* – tato data se pak načítají až po zobrazení formuláře, kdy již uživatel může zobrazený záznam editovat.

Definice formuláře Knihovna *FMForms* poskytuje jednotlivé ovládací prvky (např. textové pole, rozvinovací seznam s výběrem možností – pevně definovaných či načítaných z databáze, pole pro zadání čísla, pole kontrolované regulárním výrazem apod.). Pomocí těchto prvků lze nadefinovat formulář, sloužící k editaci jednoho řádku z tabulky (či odpovídající entity z jiného datového zdroje).

Do tohoto formuláře však lze (mimo další prvky) přidat i podřízený formulář sloužící k editaci záznamů jiné tabulky (většinou spojené s „hlavní“ tabulkou pomocí relace). V knihovně existují prvky pro provázání jednotlivých formulářů (např. „podřízený“ formulář potřebuje znát primární klíč záznamu z „hlavního“ formuláře). Tyto formuláře se načítají a ukládají z databáze pomocí jednoho příkazu – přitom jsou zpracovávány v topologickém pořadí dle svých závislostí, aby každý formulář byl zpracován až v okamžiku, kdy zná všechna potřebná data.

U každého prvku formuláře lze definovat, zdali je to prvek zobrazen, zdali se načítá z databáze a zdali je do databáze ukládán při vložení a při úpravě prvku (např. informace o času založení prvku se zapisuje pouze při vyvtoření prvku, informace o uživateli, který zápis naposledy upravil se zapisuje, ale běžnému uživateli se nezobrazuje apod.). Dále je třeba označit prvky, které slouží pro vyhledání patřičného záznamu v databázi při čtení či při zápise – dále budeme takto označené prvky nazývat klíčové (pro zápis repsektive pro čtení).



Obrázek 5.2: Schéma formuláře vytvořeného pomocí knihovny FMForms a jednotlivé kroky a postupu při jeho editaci.

Mají-li všechny klíčové prvky formuláře při čtení z databáze nastavenou hodnotu (tzn. ta není `NULL`), je do formuláře načten záznam určený těmito hodnotami, v opačném případě je uživateli nabídnut záznam nový. Mají-li všechny klíčové prvky pro zápis při zápisu do databáze hodnotu, je provedena operace UPDATE (modifikace záznamu) záznamu, definovaném těmito prvky, v opačném případě je provedena operace INSERT (založení nového záznamu). Za pozornost stojí, že pro čtení a zápis mohou být označeny jiné prvky jako klíčové – u podřízeného formuláře je běžně klíčovým prvkem pro čtení cizí klíč definující relaci na nadřízenou tabulku, zatímco pro zápis bývá zpravidla klíčovým prvkem u všech formulářů primární klíč.

Pro libovolný formulář (často se tato možnost užívá u podřízeného formuláře, jehož tabulka je s hlavním formulářem v relaci $1 : n$), je možné nastavit, aby se při čtení z databáze vygenerovala pro každý záznam vyhovující podmínkám formuláře jedna kopie tohoto formuláře. Je také možné vygenerovat navíc jeden prázdný formulář, ten pak umožní přidání nového záznamu. Bude-li tento formulář editován, vytvoří nejprve (pomocí javascriptu) svoji kopii. Tím knihovna umožňuje přidání neomezeného počtu nových záznamů stejného typu pomocí jednoho formuláře. Jednotlivé záznamy lze i označit k smazání.

Formulář se definuje pomocí zápisu kódu v *PHP* (podobně jako např. definice GUI[24] v jazyce *C#*[25]). Oproti speciálnímu jazyku na definici GUI jako např. užívá jazyk *Delphi*`http://www.codegear.com/products/delphi/win32`, byl tento způsob zvolen proto, že umožňuje snadnou variabilitu v kódu (např. definici – edituji-li heslo slovníku, nezahrnuj do formuláře podformulář pro zadávání autorství daného článku). Zároveň je tento způsob vytváření formuláře (pro uživatele knihovny) dostatečně jednoduchý. V případě potřeby (např. kvůli bezpečnosti, pokud by definice formulářů byly ukládány do databáze) není problém implementovat generování formulářů definovaných pomocí speciálního jazyka (např. na bázi *XML*).

Kontrola dat a práva Pro jednotlivé prvky formuláře lze definovat způsob kontroly zadaných dat (např. regulárním výrazem), popř. kontroly na potenciálně chybné (podezřelé) záznamy. Knihovna pak dle schématu 4.2 na straně 40 zajistí správnost zapsaných dat. Tyto kontroly se provádějí jak v klientovi pomocí javascriptu, tak i na serveru.

Pokud záznam nevyhoví kontrolám v klientovi, není povoleno odeslání formuláře. Pokud nevyhoví kontrolám na serveru (tam jsou zpravidla prováděny ještě další, komplexnější kontroly typu kontrola duplicity či jiné kolize s ostatními záznamy v databázi), místo provedení zápisu (akce ⑥ na schématu schéma 5.2 na straně 59) knihovna upozorní uživatele a vrátí se k akci ④.

Ve formuláři lze označit prvky, které jsou užity ke kontrole duplicity – existuje-li alespoň jeden takový prvek a v databázi je záznam, který se shoduje všemi takto označenými prvky, považuje se tento záznam za duplicitní. Existenci duplicitního záznamu lze buďto zakázat – knihovna nepovolí takovýto záznam uložit, nebo přikázat využití duplicitního záznamu – v tom případě jsou tyto dva záznamy sloučeny v jeden.

Ve formuláři lze také nadefinovat omezení přístupu k editovanému záznamu. Při zobrazení formuláře se tento nejprve zavolá uživatelem definovanou metodu, která formuláři vrátí práva současného uživatele k záznamu. Formulář pak místo zobrazení formuláře k editaci může pouze záznam zobrazit, či zcela odepřít uživateli přístup k záznamu.

[24] Graphical user interface, `http://en.wikipedia.org/wiki/Graphical_user_interface`

[25] `http://www.ecma-international.org/publications/files/ecma-st/ECMA-334.pdf`

Stejná kontrola se provádí i před uložením záznamu. Tato kontrola je prováděna na úrovni formuláře a nikoli *Readeru* a *Writeru*, aby bylo možné jednotlivým formulářům, přistupujícím ke stejným tabulkám, definovat různé úrovně oprávnění.

Objekty *Director* a *Event* K interakci formulářů s okolím slouží objekty *director* a *event*. Chce-li uživatel knihovny provázat formulář s dalšími objekty v aplikaci, může použít objekty v knihovně připravené, nebo napsat objekty vlastní. Jako *director* či *event* může sloužit libovolný objekt implementují rozhraní (`IFMDirector` respektive `IFMEvent`).

Pokud chce uživatel využít služby objektů *director*, musí každému typyu záznamu přiřadit objekt typu *director* (více typů však může sdílet jeden director). Pomocí těchto objektů definuje, jaký formulář se má zobrazit při zvolení daného typu záznamu a zároveň se správně inicializuje formulář (např. se při editaci záznamu před načtením prvku z databáze správně nastaví *klíčové prvky* formuláře). Jsou-li definovány tyto objekty, je následné užití knihovny velmi jednoduché: stačí knihovně přikázat: „zobraz formulář, příslušný k tomuto objektu“.

Pomocí objektu *event* lze definovat akce, které proběhnou při zobrazení formuláře či jeho zapsání do databáze (např. přidání záznamu do stromu).

Předdefinované objekty Součástí knihovny jsou objekty *director* a *event* pro editaci záznamů, které jsou zobrazeny pomocí widgetu typu strom (ne nutně víceúrovňového, tzn. je možné použít strom jako list).

Tyto předdefinované objekty *director* lze uspořádat do stromu odpovídajícímu zobrazenému stromu editovaných objektů, jeden director však lze užít pro více úrovní nebo uzlů původního stromu Např. jak zvolení záznamu typu časopis i typu desetiletí (rozmezí ročníků) má být zobrazen formulář pro přidání ročníku – tyto dva typy uzly stromu tedy sdílí jeden *director*.

Vztah mezi uzly stromu *directorů* definuje, jaké podřízené záznamy lze v daném uzlu vytvořit (to je definováno podřízenými *directory*), popř. jaký záznam má být zvolen jako aktivní po smazání záznamu (nadřízeným *directorem*). U každého *directoru* lze definovat zdali se při zvolení záznamu odpovídajícího typu má zobrazit formulář k editaci tohoto záznamu, k založení nového záznamu stejného typu, nebo k založení záznamu podřízeného.

Objekty *event* pro widget strom pak zajišťují automatické přidávání, mazání a aktualizaci prvků ve stromu podle provedených změn.

Vzhled formuláře a objekt *Renderer* Vzhled formuláře lze modifikovat pomocí kaskádových stylů. Pokud je pro uživatele tato možnost nedostatečná, je možné modifikovat (či užít vlastní) objekt *Renderer*. Tento objekt má definovány jednoduché primitivní grafické operace typu *začni skupinu záznamů*, *začni vykreslovat prvek*, *vykresli popis prvku*,*vykresli samotný prvek* apod. Dále tento objekt nabízí jednotlivým prvkům názvy css tříd, které mohou tyto objekty použít. Tyto třídy lze za běhu modifikovat a tak např. vizuálně odlišit vnořený formulář. Prvky formuláře tedy nevykreslují přímo, nýbrž používají tyto metody .

Tento koncept knihovny umožňuje snadné vizuální sladění prvků formuláře. Zároveň je tak dosaženo vysoce variabilního vzhledu webových formulářů, aniž by bylo třeba pro

odlišný vzhled formuláře modifikovat samotné ovládací prvky.

Objekty *Reader* a *Writer* Tyto objekty slouží k načítání a ukládání dat z databáze popř. jiných datových zdrojů. Jde o další vrstvu abstrakce nad abstraktní SQL vrstvou, zatímco abstraktní SQL vrstva požaduje jako svůj backend libovolný zdroj dat umožňující SQL dotazy, tyto objekty jsou ještě obecnější. Jediným jejich požadavkem je, aby datový zdroj, který užívají, měl strukturu datových tabulek – tedy uměl ukládat soubory entit s definovanými vlastnostmi a dle těchto vlastností entity vyhledávat.

Definicí patřičného objektu *Reader* a *Writer* je tedy možné užít jako datový zdroj pro knihovnu *FMForms* nejen relační databáze, ale i *XML* soubory, *csv* soubory, nebo např. formulář propojit s jinou webovou službou apod.

Variabilita knihovny Knihovna umožňuje definici libovolných formulářů. Zároveň je vysoce variabilní, takže jednoduchou úpravou či definicí nových objektů lze definovat další ovládací prvky, upravovat vzhled vygenerovaných formulářů či data ukládat do libovolných datových zdrojů. Je kompatibilní s libovolným prohlížečem podporujícím *HTML 4.0*, zároveň však umožňuje využít i skriptování na straně klienta či technologie *AHAH*. Má tedy všechny předpoklady pro užití i v dalších projektech. [26].

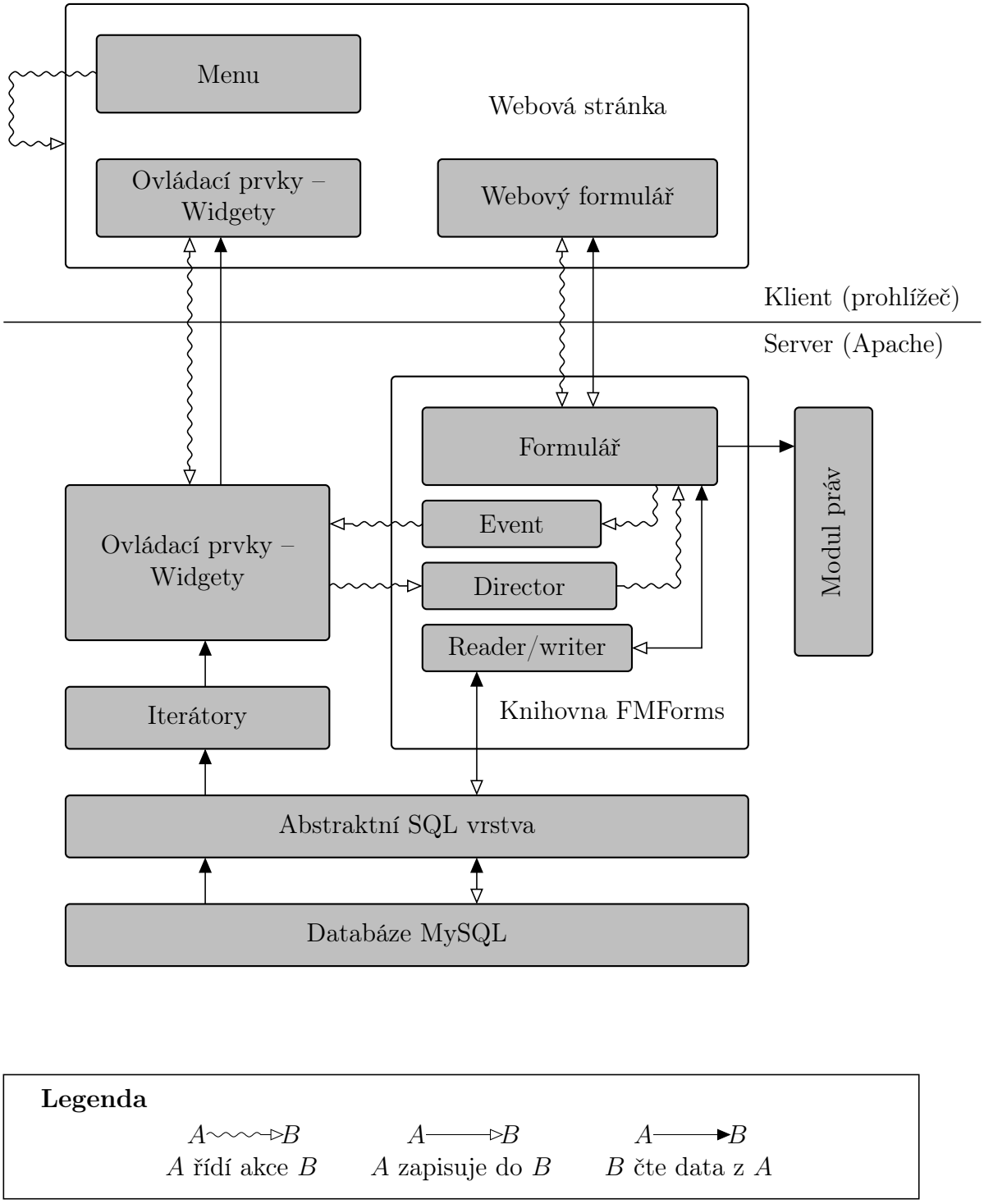
Komponenty aplikace pro zadávání

Samotná aplikace pro zadávání, jejíž schéma je na schématu 5.3 na straně 63 se pak sestává z následujících komponent:

- Iterátory, enumerující jednotlivé zobrazené objekty. Tyto iterátory zároveň poskytují o enumerovaných objektech další potřebné informace.
- *Widget* typu strom, zobrazující obsah iterátorů.
- Objekty *director*, které zajišťují zobrazení správného formuláře a *event*, které upravují strom dle provedených změn.
- Samotný objekt formuláře, zajišťující vygenerování webového formuláře na stránku. Aby se při odeslání formuláře na server nemusela obnovovat celá stránka, je tento formulář umístěn do *HTML* elementu `IFRAME`. Zbylé části stránky (mimo tento `IFRAME`) jsou upravovány pomocí javascriptu vygenerovaného objektem *event*.
- Knihovny pro přístup do databáze.
- Modul práv. Tohto modulu se formulář dotazuje, zdali má uživatel právo zobrazit, vytvořit či editovat daný záznam.
- Další komponenty: menu pro vstup do jednotlivých podaplikací, formulář na vyhledávání a rozhraní pro spouštění administračních úloh. Mezi tyto úlohy spadá např. vygenrování kódů pro prvky stromu (viz kapitola 5.2 na straně 51). Podrobněji je o této části aplikace pojednáváno v dokumentaci k systému *Depozitum*.

Samotný kód aplikace pak pouze vytváří instance výše uvedených prvků a logické vazby mezi nimi. Struktura těchto prvků je zachycena na schématu. Zbylé detaily se shodují s aplikací pro prohlížení dat a jsou zmíněné v následující kapitole. 5.3 na straně 63.

[26] Již byla např. užita v internetové anketě mezinárodního vědeckého časopisu *Studia Neoaristotelica* (`http://agora.metaphysica.skaut.org/sn/`)



Obrázek 5.3: Schéma aplikace na vytváření rejstříků

5.3.3 Aplikace sloužící k zveřejnění digitalizovaných dokumentů

K zveřejnění dat slouží dvě verze ovládacího rozhraní. Ty, ačkoli navenek vypadají naprosto odlišně, používají stejné jádro – iterátory enumerující záznamy v databázi – a pouze volí jiné prostředky k prezentaci dat tímto jádrem. Zatímco rozšířené ovládací rozhraní používá k prezentaci dat knihovnu *Widget*, jednoduché rozhraní používá knihovnu *TXGame*.

Knihovna *TXGame*

Knihovna *TXGame* slouží ke generování *příkazově orientovaného* ovládacího rozhraní. Na ovládací rozhraní poskytované touto knihovnou lze hledět jako na orientovaný graf, kdy uzly grafu jsou jednotlivé webové stránky a hrany odkazy mezi stránkami. Každý vrchol má přiřazen svůj typ, kterým je identifikován (např. zobrazení obsahu časopisu, zobrazení obsahu ročníku, vyhledávání apod.). Při přechodu mezi uzly lze však předávat navíc další údaje (např. `id` prohlíženého článku apod.).

Základním ovládacím prvkem této knihovny je zde opět *director* – který má analogickou funkci jako *director* u knihovny *FMForms*. *Director* na základě předaného typu uzlu, ve kterém se uživatel nachází, vybere objekt *textgame*, který vygeneruje uživateli obsah navštívené stránky a navigaci: odkazy na okolní stránky v grafu (a případně odkazy na předchozí navštívené stránky). Např. v případě prohlížení ročníku dostane uživatel možnost prohlížet jednotlivá čísla, možnost vyhledávat, nebo možnost vrátit se zpět k výběru titulu.

Knihovna *TXGame* nabízí standardní implementaci objektu *textgame* pro generování možnosti daných patřičným rekurzivním iterátorem (tedy iterátorem enumerujícím položky stromu[27]). Zároveň tato standardní implementace uživateli nabídne i návrat na všechny nadřízené úrovně daného iterátoru.

O samotné zobrazení zobrazení se pak stará opět objekt *renderer*. Ten nabízí metody typu začni nadpis, ukonči nadpis, začni seznam, ukonči seznam atd., což spolu s možností definovat vlastní kaskádové styly nabízí velmi rozsáhlé možnosti v konfiguraci výsledného vzhledu aplikace.

Iterátory a realizace aplikace

Jak vyplývá z návrhu specifikace v kapitole 4.4 na straně 38, jednotlivé moduly ovládacího rozhraní (prohlížení dle titulů, autorů...) lze popsat jako strom znázorněný na schématu 4.3 na straně 44. Podle tohoto schématu jsou v jádře aplikace definovány iterátory, které sdílejí obě aplikační rozhraní (a jejich mírně upravené verze užívá i aplikace pro zadávání dat). Tyto iterátory jsou pak zobrazené pomocí knihovny *TXGame*, respektive pomocí *widgetu* typu strom.

Zatímco u základního ovládacího rozhraní pokyn uživatele k zobrazení článku zobrazí samostatné okno, ve kterém jsou zobrazeny informace o článku a jeho jednotlivé stránky; v případě rozšířeného ovládacího rozhraní je vedle stromu s svazky, klíčovými slovy či autory a články rovnou zobrazován obsah článku. V rozšířeném rozhraní samozřejmě nechybí možnost otevřít daný článek v samostatném okně.

[27] `http://www.php.net/~helly/php/ext/spl/interfaceRecursiveIterator.html`

Kapitola 6

Zkušenosti s projektem

V této kapitole se budeme věnovat zkušenostem získaným během pilotního projektu a některým z toho plynoucím rozhodnutím. Nyní se budeme podrobněji věnovat jednotlivým částem prováděné digitalizace.

6.1 Scanování

Samotné scanování probíhalo na dvou typech scanerů. Většina materiálu byla digitalizována pomocí scanneru *Plustek book reader*[1]. Tento scanner je však schopen pojmout předlohy maximálně velikosti A4, proto pro větší předlohy zakoupila Katolická teologická fakulta scanner *Avision FB6080E*[2], schopný zpracovávat předlohy do velikosti A3. Tento scanner ještě nebyl do práce na projektu nasazen a jeho plné možnosti se v současné době testují. Oba tyto scanery jsou specializované na digitalizaci knih (plocha pro předlohu umístěná nahoře dosahuje na jedné straně až k boku přístroje).

Věžný postup scanování bylo takzvané *dávkové scanování* – mód scanování, kdy je scanována jedna stránka za druhou a jsou automaticky pojmenovávány. Tvar vytvořeného souboru je `<prefix><číslo>`, kde `<prefix>` je nastaven uživatelem a číslo (z počátku taktéž uživatelem nastaveno) se při každé nascanované stránce zvedne o jedna. Tento mód nabízí jak ovládací software pro scanner, tak i některý běžně dostupný software (např. IrfanView) plně vyhovuje zavedenému způsobu číslování stran (viz kapitola 4.2.2 na straně 31) – ruční zásah do automaticky generované sekvence je třeba pouze pokud jsou v dokumentu nepravidelnosti v číslování či strany bez natisknutého čísla (pokud jich je více za sebou, pak pouze pro první stranu, neboť i zde lze nastavit vhodný prefix a stany se budou číslovat správně).

Z přímých porovnání těchto scanerů plynou následující pozorování

- Rychlost scannování na scanneru Plustec dosahoval u zapracovaného člověka průměrné rychlosti 100 stran za hodinu.
- Scanner Plustec je o něco rychlejší než Avision.
- Scanner Avision má funkci automatické detekce scanovaného materiálu na ploše. U scaneru Plustec je třeba manuálně nastavit ořez stránky. Při dávkovém scanování lze tento ořez nastavit při hraně scaneru, což umožňuje scanování neomezeně strá-

[1] `http://www.plustek.com/bookreader/index.asp`
[2] `http://www.avision.com/usa/products/fb6080e.html`

nek na jedno nastavení ořezu, není to však tak přesné jako automatická detekce, někdy dochází k nechtěnému ořezu stránky.

- Scanner Plustec má ve svém ovládacím programu možnost každou sudou stránku otočit o 180°. To je třeba, neboť hrana, ke které je možné přiložit hřbet knihy je jen na jedné straně scanneru. Absence této možnosti u scanneru Avision znemožňuje přímému použití jeho ovládacího programu pro dávkové scanování, místo toho je třeba použít externí utilitu (např. zmíněný IrfanView).
- Nascanované stránky nejsou vždy zcela správně otočeny (svislé hrany tedy nejsou někdy zcela svislé). Automatickou korekci rotace (dle detekce hran – v tištěném textu by měly převažovat vodorovné a kolmé hrany) nenabízí ani jeden z ovládacích programů ke scannerům. Vzhledem k malým rotacím (chyba dosahuje výjměčně dosahují 10°, většina digitalizovaného materiálu má svislice v pořádku) nebyl tento problém řešen.

Z výše vypsaných pozorování vyplývá, že pro současné potřeby a rozměry digitalizace není třeba vyvíjet specializovaný nástroj na scanování, pokud by byla započata digitalizace ve velkém měřítku, bylo by možné uvažovat o specializovaném programu, který by eliminoval nevýhody ovládacích programů jednotlivých scannerů a nabízel by další přidané funkce (automatické číslování stran dle užívaného formátu, detekce strany a korekce natočení apod.)

Zmenšené kopie digitalizovaných stran se generují pomocí dávkového zpracování pomocí grafického software IrfanView.

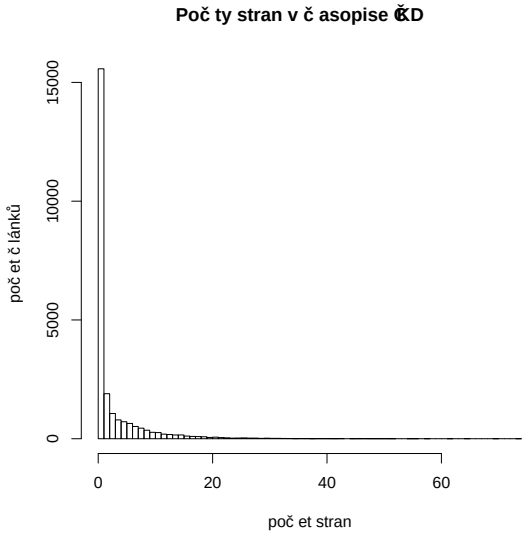
Velikost uložených obrázků záleží na použitém kódování u *TIFF* souborů. U souboru velikost 1275×2219 je velikost originálního souboru $8.5MB$ u kódování packedbits, $7.4MB$ u kódování *LZW* a $6.3MB$ s kódováním *ZIP*. Zmenšený JPEG má velikost $250kB$ až $400kB$.

6.2 Pořizování rejstříků

Pořizování rejstříků je zatím nejnáročnější prováděná část celého projektu. Testované automatické zpracovávání obsahů se neukázalo rychlejší než ruční zpracování. Příprava strany převedené pomocí OCR do textu zabrala příliš dlouho, bylo nutné provést textovou korekturu, dále neexistoval jednotný formát pro sazbu obsahu a tak musel být každý obsah ručně rozdělen na patřičné sloupce. To, spolu s faktem, že v obsahu chybějí některá data (nekompletní jména autorů, chybějící koncové strany článků atd. . .) vedlo k tomu, že byla myšlenka automatického generování rejstříků prozatím opuštěna.

Stručná statistická data o časopisech Počet článků na jedno číslo je cca 26 (*Český časopis historický* - dále ČČH), 178 (*Časopis Katolického Duchovenstva*, dále ČKD) respektive 851 u Latinskho slovníku (dále slovník). V ČČH připadají tři strany na jeden článek, u ČKD 1, 5 strany na článek, u Latinského slovníku připadá cca 10 článků (hesel) na stránku. Pokud budeme sledovat rozsah jednotlivých (základních) článků (strana je k článku započítána vždy celá, nezávisle na tom, zdali ji článek zabírá celou či ji sdílí s jiným článkem), získáme u ČKD tyto data: medián 2, průměr 3.89, směrodatná odchylka 5.28. Histogram stranového rozsahu článků je zachycen na obrázku 6.1.[3]

[3]Tato čísla jsou větší než v předešlém odstavci, neboť na některých stranách je více krátkých článků.



Obrázek 6.1: Histogram stranového rozsahu článků v Časopise katolického duchovenstva

Výkonnost pracovníků Při ručním pořizování rejstříků se ukázalo naprostou nutností hodnotit pracovníky (včetně pracovníků se stálým úvazkem) nikoli „od hodiny“, nýbrž „od záznamu“ – produktivita práce u testovaných osob tak velmi vzrostla (skoro trojnásobně). Po provedení prvotních testů byla pro potřeby finančního ohodnocení pracovníků stanovena norma 160 sekund na pořízení jednoho záznamu. Podle pozdějších analýzy pracovních výsledků byl však tento limit u zkušených pracovníků poněkud podhodnocen.

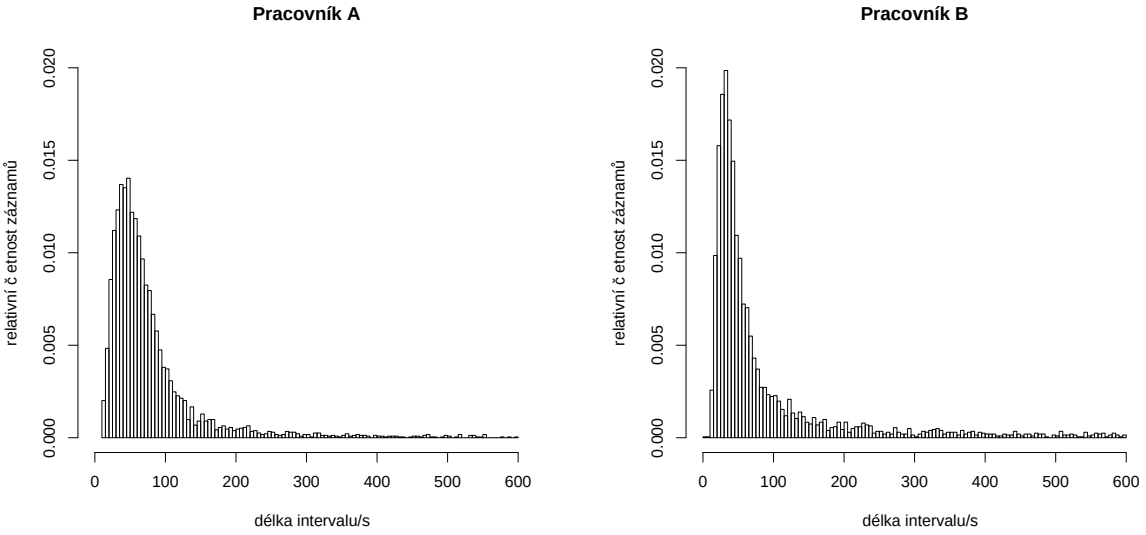
Byly sledováni dva pracovníci, oba byly schopní rychlé práce na počítači.[4] Čas na zpracování článku byl měřen jako intervaly mezi dvěma zadanými záznamy, přičemž bereme v úvahu pouze intervaly kratší deseti minut. Průměrný čas na jeden záznam u pracovníka A 85 sekund (celkem 4.676 záznamů) a pracovníka B 78 sekund (celkem 4040 záznamů) při mediánu 59 respektive 46 sekund a směrodatné odchylce 71 respektive 104. Rozložení jednotlivých intervalů korespondovalo s očekávaným exponenciálním rozdělením (viz obrázek 6.2).

Rozdíl mezi pracovníky je pravděpodobně způsoben jazykovou odlišností zpracovávaného dokumentu. Pracovník A během sledované doby pracoval na časopise ČKD z devatenáctého století, který má nezanedbatelnou část ročníků psanou starým pravopisem (dlouhé ‚í’ se psalo jako ‚j’, ‚j’ jako ‚g’ apod.) . Pracovník B zpracovával ČČH z počátku dvacátého století, který je psán pravopisem velmi blízkým dnešnímu. V svazcích přidělených pracovníkovi B bylo více cizojazyčných názvů článků než ve svazcích zpracovávaných pracovníkem A (cca 30% ku 5%).

Omezíme-li sledované záznamy u pracovníka A na ročníky ČKD psané nejstarším typem pravopisu, dosahuje pracovník A průměrného času 103 sekund na záznam (při počtu 1.291 záznamů, mediánu 84 sekund a směrodatné odchylce 74). Naopak v ročnících ČKD z počátku dvacátého století, které jsou již psány (až na drobné odchylky) dnešním pravopisem, dosáhl pracovník A průměrného času 60 sekund na jeden záznam (3.910 záznamů, medián 46 sekund a směrodatná odchylka 60).

Za povšimnutí stojí, že nejkratší dosahovaný čas na jeden záznam (dosahovaný u

[4]Všechna sledovaná data byla pořízena po dostatečném zaškolení pracovníků.



Obrázek 6.2: Histogram relativní četnosti intervalů mezi zadáním jednotlivých záznamů u pracovníků pořizujících rejstřík. Histogram je členěn do skupin po třech sekundách, na ose Y je vynesen poměr *počet článků v dané skupině/počet všech článků.*

velmi jednoduchých záznamů typu „Obsah“) je u pracovníka A 11 sekund, zatímco u pracovníka B 12 sekund, je tedy nepravděpodobné, že by pracovník A byl méně výkonný než pracovník B, ten navíc dle častějších výskytů delších intervalů nebyl schopen pracovat tak soustředěně jako A.

Získaná data ukazují, že čeština psaná starým pravopisem je na přepis podobně obtížná jako články v cizím (částečně ovládaném) jazyce: rychlost zpracování časopisu s částí německých titulků je někde mezi rychlostí zpracování čaospisu českého psaného dnešním respektive historickým pravopisem.

Při zpracovávání ČČH se lze také ukázat, že při zpracovávání cizojazyčných článků je důležitá jazyková vybavenost: jazykově velmi dobře vybavený pracovník C byl schopen zpracovávat ČČH s průměrným časem 54 sekund na záznam (3933 záznamů, medián 42 sekund, směrodatná odchylka 51), zatímco špatně jazykově vybavený pracovník D dosahoval času 90 sekund (3334 záznamů, medián 75, směrodatná odchylka 62). To přibližně koresponduje s výkonem pracovníka A na dokumentech psaných starou češtinou, přihlédneme-li k tomu, že pracovník A měl většinu záznamů psanou „cizím“ historickým pravopisem, zatímco D zpracovával směs cizojazyčných titulků (pro něj však obtížněji čitelných) a českých titulků.

Předpokládaná hypotéza závislosti délky článku (a tedy nutnosti prohlédnout větší část dokumentu na zapsání jednoho článku) na rychlosti zpracování se nepotvrdila. Přestože ČČH má delší články než ČKD, pracovník C ho byl schopen zpracovávat rychleji, než pracovník A časopis ČKD. Pravděpodobně přinejmenším stejně významným faktorem je větší přehlednost časopisu (delší články jsou zpravidla výrazněji odlišeny).

U kratších intervalů ze vysledovat korelaci mezi délkou názvu článku a dobou pořízení záznamu. U souboru 3.565 záznamů pořizovaných kratší dobu než dvě minuty je Pearsonův korelační koeficient 0.68.

Ze získaných dat lze odhadnout, že zaučený pracovník schopný rychlé práce na počítači je schopen pořídit průměrně jeden záznam za přibližně 50 až 60 sekund, u do-

kumentu psaném cizím jazykem je čas zhruba dvojnásobný. Za cizí jazyk lze pro tento účel považovat i starou češtinu,[5] naopak jazykově vybavený pracovník je schopen zaznamenávat cizojazyčné záznamy v podobné rychlosti, jako svůj rodný jazyk. Tomuto odhadu odpovídají i dosahované časy u ČČH, který obsahuje směs českých a cizojazyčných článků. Tento odhad je však velmi orientační: jak vyplývá z pozorování, i v rámci jednoho dokumentu se mohou časy různých pracovníků lišit (dle schopnosti pracovníka i aktuální náročnosti daného svazku).

Jako u každé činnosti, tak i u zpracovávání rejstříků je třeba počítat s určitým časem pro zaškolení pracovníka. Ukazuje se, že čas pracovníka na zpracování jednoho záznamu se ustálí po pořízení přibližně tisíce záznamů. Zadávání rejstříků je vhodné kombinovat s jinou činností, neboť je to práce poměrně úmorná a málokterý pracovník ji vydržel vykonávat souvisle delší dobu než několik málo hodin.

Získané odhady by šlo rozšířit a zpřesnit detailnější analýzou záznamů, v které by se oddělili české a cizojazyčné články, analýzou závislosti rychlosti zpracování záznamu na délce zadaných dat, popř. měřením výkonnosti pracovníků na vzorových množinách různě obtížných dat. Pro stanovení odhadu délky zpracování rejstříků a pro účely ohodnocení pracovníků je však tato analýza dostatečná.

Při sběru údajů se také neměřila přesně chybovost pracovníků; obecně se však dá říci, že pořízené rejstříky nejsou zcela bez chyby, tato chybovost však byla vzhledem k množství záznamů zanedbatelná a nebylo třeba ji speciálně řešit.

Další zkušenosti Při pořizování rejstříků se obzvláště osvědčilo využití pracoviště s dvěma monitory, na jednom je zobrazena samotná aplikace, na druhém pak v prohlížeči obrázků digitalizované strany. Po zaškolení pracovníků byly rejstříky často pořizovány pomocí internetu z pracovišť mimo samotnou fakultu.

V pořízených rejstřících nejsou výjimkou chyby, některé (např. překrývající se články, články s rozsahem přes více čísel) lze odchytit skripty, které zároveň zpracovávají scanované soubory stran a přidávají je do databáze, některé (obzvláště překlepy) však automaticky odchytit nelze. Jelikož detailní kontrola by byla příliš nákladná, publikují se digitalizované dokumenty i s případnými chybami, přičemž se spoléhá na součinnost uživatelů systému při jejich odstraňování.

6.3 Další zkušenosti s digitalizací dokumentů

Zbylé procesy (mimo kontroly dat, prováděné namátkou pověřeným pracovníkem) tvoří pouze jednorázové akce, které nejsou příliš zajímavé z technologické ani ekonomické stránky. Jedná se o přidání nascanovaných stran do databáze a zveřejnění dokončeného titulu. Nascanované strany jsou do databáze (respektive do datového úložiště a odkazy na ně do databáze) přidávány pomocí skriptu, další skripty existují pro vytvoření stromových struktur v databázi (viz kapitola 5.2 na straně 51) a další jednorázové činnosti. Při těchto činnostech jsme nenarazili na žádný větší problém ani technologickou obtíž, kterou by bylo třeba zmínit.

Zveřejnění digitalizovaných dokumentů je jednoduchý proces, kdy je databáze, zatím zobrazovaná pouze v interní verzi aplikace, přidána mezi databáze zobrazované ve

[5] S přihlédnutím k tomu, že stará čeština je pro čecha srozumitelná, rychlost jejího zpracování tedy bude vyšší než u zcela neznámého jazyka.

veřejné verzi, případně je titul v již zveřejněné databázi označen jako publikovaný. I tato činnost s sebou nenese žádné problémy, které by byly hodné zřetele.

Aplikace pro zadávání dat slouží svému účelu a je používána mnoha lidmi, o čemž svědčí i počet a hlavně pořízených a rychlost pořizování záznamů – doposud bylo pořízeno cca 100 000 záznamů typu článek. Jelikož je již započatá práce na několika typech digitalizovaných dokumentů, osvědčila se i modulárnost této aplikace (díky použité knihovně *FMForms*).

Očekávání, vkládané do aplikace pro zveřejnění byly taktéž splněny: zatímco jednoduché uživatelské rozhraní je velmi blízké začínajícím uživatelům a je široce kladně hodnoceno, pokročilí uživatelé systému často volí rozšířené rozhraní, v kterém je na úkor přehlednosti možné některé činnosti vykonávat rychleji.

6.4 Ekonomická náročnost projektu

Nyní provedeme kalkulaci ceny základní digitalizace typického časopisu. Půjde o velice hrubou kalkulaci, zahrnující pouze přímé náklady s digitalizací tohoto časopisu, bez započítání samotné ceny na vývoj řešení, cenu hardwarového vybavení, používaného k digitalizaci a běhu potřebného software, plat lidí starajících se o tento software a hardware atd, neboť s přibývajícím počtem digitalizovaných dokumentů se bude se bude podíl těchto nákladů připadajících na jeden dokument snižovat. Zároveň kalkuluje s relativně nejnižšími možnými cenami lidské práce (brigádníci apod.). Také Další fáze digitalizace, kdy je zapotřebí kvalifikované pracovní síly, by byly ještě dražší. Jde tedy spíše o stanovení minimální ceny za samotnou digitalizaci, zhruba stanovující rámec ceny digitalizace jednoho titulu.

Na digitalizaci jednoho časopisu, o rozsahu cca 100 ročníků po dvanácti číslech, se sedmdesáti stranami na číslo (zaokrouhlený rozsah *Časopisu katolického duchovenstva*, celkem 84.000 stran) a třemi stránkami na článek jsou zapotřebí následující zdroje:

Diskový prostor 3×750 (originál + zálohy) (dva desktopové SATA disky + externí disk)	8 000,− Kč
Plat lidí pro nascanování digitálních dat (100,−Kč/hodinu, 100 stran/hodinu)	84 000,− Kč
Plat lidí pro vytvoření rejstříku (100,− Kč/hodinu, 90 s/záznam, 3 strany/záznam)	70 000,− Kč
Celkem	162 000,− Kč

Z vypočtených nákladů je vidět, že zdaleka největší položkou v ceně digitalizace je lidská práce, která dosahuje ceny cca 2,− Kč za stranu. Ta je v současné době prakticky neodstranitelná. Z tohoto důvodu zatím nebylo zatím přistoupeno (ačkoli je k tomu připraveno či skoro připraveno aplikační rozhraní) ke korekturám a zveřejnění textové podoby článků.

Tato kalkulace je velmi orientační, neboť rychlost pořizování rejstříků velmi kolísá, nejen jak jsme ukázali kvůli různé náročnosti přepisování názvů článků, ale také díky odlišnému poměru stránek k článkům. Proto např. u latinského slovníku je díky mnoha krátkým heslům cena za pořízení rejstříku přibližně třikrát větší, než cena samotného pořízení digitálních kopií.

6.5 Zkušenosti uživatelů s projektem *Depozitum*

V reakci na pilotní projekt jsme zaznamenali reakci uživatelů z několika univerzit: mimo samotné *Katolické teologické fakulty* máme kladné ohlasy mimo jiné z *Teologické fakulty Jihočeské univerzity*[6], *Cyrilometodějské teologické fakulty Olomoucké univerzity*[7] či *Evangelické teologické fakulty.*[8] Zcela dle očekávání vyhovuje začínajícím uživatelům jednoduché uživatelské rozhraní, existují však zběhlí uživatelé, kteří preferují rozhraní pokročilé.

Projekt *Depozitum* také zaujal některé další subjekty, které projevily zájem o vyvinutou technologii, např. *Ústav pro studium totalitních režimů.*[9] S některými subjekty je již navázaná spolupráce, např. s *Knihovnou samizdatové a exilové literatury Libri Prohibiti*[10] se kterou spolupracujeme na digitalizaci časopisu *Život*, v současné době také probíhá jednání o spolupráci s *Akademií věd České republiky*[11] na dokončení digitalizace *Časopisu katolického duchovenstva.*

6.6 Další rozvoj projektu

Problémem projektu, nastíněným v kapitole 6.4 na straně 70 je vysoká finanční náročnost digitalizace. Ta zatím brání většímu rozšíření projektu, ať již jde o množství digitalizovaných titulů, tak i o využití technologií (např. převádění dokumentů do textové podoby, případná lokalizace přesné pozice vyhledávaného textu na stránce apod.).

V současné době čeká projekt rozšíření o další digitalizované tituly. Po zveřejnění více titulů je třeba také provést audit výkonnosti jednotlivých komponent systému, monitorovat výkonnost SQL dotazů a popřípadě doplnit potřebné indexy. V současné době také probíhá převod prvních digitalizovaných dokumentů do textové podoby, což umožní dokončit zpřístupnit dokumenty pomocí fulltextového prohledávání.

V technologické oblasti je třeba sesbírat dlouhodobější reakce uživatelů na uživatelské prostředí a provést případné úpravy dle požadavků, je možné uvažovat i o implementaci silnějších vyhledávacích nástrojů. Při současné klesající ceně pevných disků je také možné uvažovat o zvýšení kvality pořizovaných kopií, je však třeba počítat i s možnými vyššími nároky na finanční náročnost digitalizace, neboť pořizování digitálních kopií ve vyšším rozlišení prodlouží dobu scanování jedné stránky. Při dalším rozvoji projektu by pak bylo velmi vhodné zavést katalogizaci titulů pomocí vhodného formátu a propojení s dalšími knihovními systémy.

[6] `http://www.tf.jcu.cz/`
[7] `http://www.upol.cz/fakulty/cmtf/`
[8] `http://www.etf.cuni.cz`
[9] `http://www.ustrcr.cz/`
[10] `http://libpro.cts.cuni.cz/`
[11] `http://www.avcr.cz/`

Kapitola 7

Závěr

V pilotním provozu projektu se podařilo s úspěchem vyzkoušet navržené technologie a digitalizovat několik velmi rozsáhlých dokumentů (např. *Časopis katolického duchovenstva* má přes 77.000 stran). V současné době je dokončena digitalizace čtyř titulů,[1] ovšem množství dalších titulů je rozpracováno v některém ze stádií digitalizace.

Množství pořízených dat v databázi i celková velikost digitálních kopií dokumentů, která dosahuje skoro tří terabytů i zisk grantů na provoz tohoto projektu, to vše svědčí o jeho životaschopnosti. Tento fakt potvrzuje i zájem dalších subjektů o spolupráci na tomto projektu.

Během projektu zároveň vznikly i některé technologie, které nejsou přímo vázány na tento projekt. Jedná se především o knihovny *FMForms* a *Widget*, které jsou zcela univerzální a tak mohou být použity v libovolném dalším projektu. Obě tyto knihovny již také byly použity při vývoji jiných aplikací, např. jako součást komplexního řešení zálohování dat ve *Fyzikálním ústavu Akademie věd České Republiky*,[2] nebo technické řešení internetové průzkumu pro mezinárodní vědecký časopis *Studia Neoaristotelica*.[3]

Máme-li stručně shrnout, co bylo výsledkem projektu *Depozitum*, můžeme konstatovat, že vznikl projekt, který může pomoci zachránit historické dokumenty pro příští generace, přičemž je zároveň uživatelsky příjemným způsobem zveřejní na internetu. To nejen umožní odborné veřejnosti daleko snažší a efektivnější vědeckou práci s těmito dokumenty, zároveň je však zpřístupní i laické veřejnosti, která by k nim jinak těžko získávala přístup.

[1] Nejsou však u nich provedeny všechny fáze digitalizace, ke všem „dokončeným“ dokumentům je však pořízen rejstřík a dokumenty jsou zveřejněny.

[2] `http://www.fzu.cz`

[3] `http://agora.metaphysica.skaut.org/sn/`

Literatura

[1] AiP Beroun. MANUSCRIPTORIUM V. 2.0, Komplexní digitální dokument, Draft, verze 1.0, 2005. http://digit.nkp.cz/projekty/VZ-2004_2010/2005/ManuscriptoriumKDD.pdf.

[2] AiP Beroun. MANUSCRIPTORIUM V. 2.0, OAI-PMH Harvester. Dokumentace k programu, 2007.

[3] AiP Beroun s.r.o. Manuscriptum Quality – Kvalita obrazových dat, 2006. http://www.manuscriptorium.com/Download/Documentation/manuscriptorium_image_quality_CZE.pdf.

[4] AiP Beroun s.r.o (Karel Kučera, František Šibrava, Martin Majer). Manuscriptorium v. 1.0 – Manuscriptorium technical compatible, 2006. http://www.manuscriptorium.com/Download/Documentation/manuscriptorium_compatibility_technical_CZE.pdf.

[5] Troels Arvin. Comparison of different SQL implementations, 2007. http://troels.arvin.dk/db/rdbms/.

[6] Marie Balíková. Předmětová kategorizace informačních zdrojů pro potřeby Konspektu. *Národní knihovna: Knihovnická revue*, 1:42–54, 2003.

[7] M. Benoit. Linux File System Benchmarks , 2003. http://fsbench.netnation.com/.

[8] Ecma International. ECMAScript Language Specification, 1999. http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-262.pdf.

[9] Daniele Florio. AHAH Section, experimental project by D.Florio, 2006. http://www.gizax.it/ahahsection/.

[10] Betty Furrie. *Understanding MARC Bibliographic: Machine-Readable Cataloging*. Cataloging Distribution Service, Library of Congress ve spolupráci s Follett Software Co. in, 2000.

[11] Diane Hillmann. Using Dublin Core, 2005. http://dublincore.org/documents/usageguide/.

[12] Olga Hándlová. Dějiny a vývoj MDT, 2004. http://www.phil.muni.cz/kivi/clanky.php?cl=25&rubrika=clanky.

[13] International ISBN Agency. ISBN Users' Manual, fifth edition, 2007. http://www.isbn-international.org/en/download/2005%20ISBN%20Users%27%20Manual%20International%20Edition.pdf.

[14] ISSN International Centre. ISSN Manual. Cataloguing Part, 2002. http://www.issn.org/files/active/0/manual.pdf.

[15] Radim Chvála Iva Horová. Zpřístupňování digitalizovaných dokumentů: lokální aktivity versus celonárodní projekt?, 2006. http://www.evskp.cz/Dokumentyver/kniznica_horova.pdf.

[16] Hans Ivers. Filesystems (ext3, reiser, xfs, jfs) comparison on Debian Etch , 2006. http://www.debian-administration.org/articles/388.

[17] Maria Inês Cordeiro Joaquim Ramos de Carvalho. XML and Bibliographic Data: the TVS (Transport, Validation and Services) Model. *68th IFLA General Conference and Council*, 2002.

[18] (AiP Beroun) Karel Kučera. MANUSCRIPTORIUM V. 2.0, Systém pro správu metadat, 2007. http://digit.nkp.cz/projekty/VZ-2004_2010/2007/Prilohy/Mns_DBEZ1_071031.pdf.

[19] Peter-Paul Koch. Speed test: innerHTML versus DOM manipulation, 2008. http://www.quirksmode.org/dom/innerhtml.html.

[20] Masarykova universita v Brně. Dublin Core Czech, 2006. http://www.ics.muni.cz/dublin_core/index.html.

[21] MASTER Work Group. Reference Manual for the MASTER Document Type Definition, Discussion Draft. http://www.tei-c.org.uk/Master/Reference/oldindex.html.

[22] National Information Standards Organization. Information Retrieval (Z39.50): Application Service Definition and Protocol Specification, 2002. http://www.loc.gov/z3950/agency/Z39-50-2003.pdf.

[23] Online Computer Library Center. Introduction to Dewey Decimal Classification, (2008). http://www.oclc.org/dewey/versions/ddc22print/intro.pdf.

[24] Jim Mostek Philip Trautman. Scalability and Performance in Modern File Systems , 1997. http://linux-xfs.sgi.com/projects/xfs/papers/xfs_white/xfs_white_paper.html.

[25] Daniel Phillips. A Directory Index for Ext2, 2001. http://www.usenix.org/publications/library/proceedings/als01/full_papers/phillips/phillips_html/index.html.

[26] J. Piszcz. Benchmarking Filesystems Part II. Linux . *Linux Gazette, 122*, January 2006.

[27] Josef Schwarz. Metoda Konspektu a její vztah k selekčním jazykům: současný stav a trendy se zaměřením na situaci v České republice. Diplomová práce, Masarykova universita v Brně, Filosofická fakulta, 2005.

[28] Text Encoding Initiative. P5: Guidelines for Electronic Text Encoding and Interchange – chapter 10: Manuscript Description. http://www.tei-c.org/release/doc/tei-p5-doc/html/MS.html.

[29] Text Encoding Initiative Consortium. TEI P5: Guidelines for Electronic Text Encoding and Interchange, 2008. http://www.tei-c.org/release/doc/tei-p5-doc/en/Guidelines.pdf.

[30] The Library of Congress. MARC Standards. http://www.loc.gov/marc/.

[31] The Library of Congress. MARCXML, Marc 21 XML Schema. http://www.loc.gov/standards/marcxml/.

[32] The Library of Congress. METS: An Overview & Tutorial, 2006. http://www.loc.gov/standards/mets/METSOverview.v2.html.

[33] The Library of Congress. <METS> Metadata Encoding and Transmission Standard: Primer and Reference Manual, version 1.6, 2007. http://www.loc.gov/standards/mets/METSmsw.pdf.

[34] The Library of Congress. METS – Metadata Encoding and Transmission Standard, 2008. http://www.loc.gov/standards/mets/.

[35] The Library of Congress. The Library of Congress Classification, 2008. http://www.loc.gov/aba/cataloging/classification/.

[36] UDC Consortium. About Universal Decimal Classification and the UDC Consortium, (2008). http://www.udcc.org/about.htm.

[37] Zdeněk Uhlíř. Projekt master a standardizace v oblasti zpracování rukopisů. *Národní knihovna: Knihovnická revue*, 3:109–113, 1999.

[38] Zdeněk Uhlíř. Standard MASTER: katalogizace rukopisů v XML. *Národní knihovna: Knihovnická revue*, 2:84–101, 2002.

[39] Zdeněk Uhlíř. Manuscriptorium na cestě k evropské digitální knihovně. *Knihovny současnosti 2007*, 2007.

[40] Helena Veličková. Klíčová slova a vybrané znaky MDT. *Knihovní obzor*, 1998.

[41] World Wide Web Consortium. HTML 4.01 Specification, 1999. http://www.w3.org/TR/REC-html40/.

[42] World Wide Web Consortium. Extensible Markup Language (XML), 2003. http://www.w3.org/XML/.

[43] World Wide Web Consortium. Document Object Model (DOM), 2005. http://www.w3.org/DOM/.

[44] Petr Zelenka, 2005. http://interval.cz/clanky/metody-ukladani-stromovych-dat-v-relacnich-databazich/.

[45] AHAH: Asynchronous HTML and HTTP, (2008). http://microformats.org/wiki/rest/ahah.

[46] http://www.php.cz, (2008).

[47] Dublin Core Metadata Initiative, (2008). http://dublincore.org/.

[48] Jednotná informační brána, (2008). http://info.jib.cz/.

[49] http://www.mysql.com/, (2008).

[50] Používat či nepoužívat mysql (diskuse), (2008). http://www.root.cz/zpravicky/pouzivat-ci-nepouzivat-mysql/.

[51] Text Encoding Initiative Consortium, (2008). http://www.tei-c.org.

Přílohy

Součástí práce je CD, na kterém lze nalézt tyto elektronické přílohy:

1. Uživatelskou dokumentaci k projektu
2. Dokumentaci pro administrátora projektu
3. Programátorskou dokumentaci k projektu
4. Samotný kód aplikace pro zadávání a zveřejnění