VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

KNIHOVNA ZDROJŮ A CITACÍ PRO DOKUWIKI

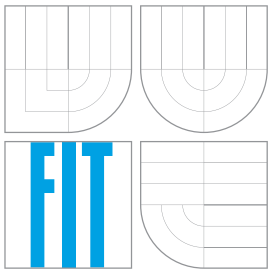
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAROSLAV KUBÍK

BRNO 2015

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

KNIHOVNA ZDROJŮ A CITACÍ PRO DOKUWIKI

RESOURCES AND CITATIONS LIBRARY FOR DOKUWIKI

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JAROSLAV KUBÍK

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. ZBYNĚK KŘIVKA, Ph.D.

BRNO 2015

Abstrakt

Cílem této bakalářské práce je návrh a implementace knihovny publikací jako zásuvného modulu pro wiki systém DokuWiki. Práce obsahuje základní informace o DokuWiki a tvorbě jejich doplňků. Dále je zde rozepsána funkcionalita, kterou bude výsledný modul obsahovat spolu s návrhem datového modelu. V poslední části je poté popsáno, jakým způsobem byly jednotlivé funkce implementovány, a jaké externí knihovny k tomu byly využity.

Abstract

The goal of this bachelor's thesis is to design and implement a resource library as a plugin for DokuWiki wiki system. This text contains basic information about DokuWiki and the plugin development. Then, the description of the final application follows along with the design of the data model. The last section covers the implementation of the application and contains informations about used external libraries.

Klíčová slova

DokuWiki, wiki, zásuvný modul, knihovna, citace, publikace

Keywords

DokuWiki, wiki, plugin, library, citation, publication

Citace

Jaroslav Kubík: Knihovna zdrojů a citací pro DokuWiki, bakalářská práce, Brno, FIT VUT v Brně, 2015

Knihovna zdrojů a citací pro DokuWiki

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Zbyňka Křivky, Ph.D.

.....
Jaroslav Kubík
20. května 2015

Poděkování

Rád bych poděkoval vedoucímu mé bakalářské práce Ing. Zbyňku Křivkovi, Ph.D. za pomoc, ochotu a čas, který mi věnoval.

Obsah

Kapitola 1

Úvod

Cílem této práce je návrh a implementace zásuvného modulu pro systém DokuWiki, který umožní katalogizovat různé publikační zdroje. Modul musí dostatečně dobře navazovat na zbytek wiki systému, a zároveň nabízet veškerou potřebnou funkcionalitu tak, aby se dal posléze v nějakém reálném systému využívat.

V úvodní kapitole 2 tohoto dokumentu je popsán samotný systém DokuWiki. Jsou zde popsány jeho nejdůležitější vlastnosti a využití. Dále tato kapitola také obsahuje základní informace o tvorbě zásuvných modulů. To zahrnuje postup vytváření takovýchto modulů a dále také detailnější popis jednotlivých typů rozšíření a ostatních funkcí, které budou poté při implementaci použity.

Kapitola 3 se zabývá návrhem našeho zásuvného modulu. Nejdříve je zde uvedena specifikace požadavků, které bude muset výsledný produkt splňovat. Dále jsou zde detailněji popsány jednotlivé problémy, které je třeba vyřešit a je zde také načrtnuto, z jakých částí se bude rozšíření skládat spolu s popisem jednotlivých funkcí těchto částí. Součástí této kapitoly je také návrh datového modelu tohoto rozšíření.

V poslední kapitole 4 je poté popisována samotná implementace rozšíření. Jsou zde detailně popsány jednotlivé třídy, včetně jejich nejdůležitějších funkcí. Jsou zde také popsány použité technologie či externí knihovny. Součástí je také zmínka o tom, jak se rozšíření testovalo.

Na závěr jsou shrnuty dosažené výsledky. Jsou zde popsány výhody a nevýhody implementovaného rozšíření a také jsou tu rozvedeny možnosti dalšího vývoje.

Kapitola 2

DokuWiki

DokuWiki [6] je jednoduchý a nenáročný Open Source wiki systém. Zaměřuje se především na tvorbu dokumentací všeho druhu. Vytváření a editování nových wiki stránek je jednoduché a intuitivní. Při tvorbě dokumentů se používá čitelná a snadno zapamatovatelná syntaxe. Tento systém také uchovává veškerou historii změn, které kdy byly provedeny. Díky tomu všemu je vhodný hlavně pro vývojové týmy či pracovní skupiny, jejímž členům umožňuje mezi sebou rychle a efektivně komunikovat a sdílet informace.

Jedná se o wiki napsanou v jazyce PHP [3], lze ji tedy bez problémů provozovat na většině běžných hostingů. Jelikož se všechna data ukládají do obyčejných textových souborů, tak ke svému provozu nepotřebuje žádný databázový server. Díky tomu se také snadněji zálohuje a uložené stránky lze případně přečíst či měnit v jakémkoli běžném textovém editoru.

Obsahuje též vestavěnou podporu pro nastavování přístupových práv. Přístup jednotlivých uživatelů nebo celých skupin lze jasně definovat pomocí pravidel. Dále také nabízí systém pro správu médií, který umožňuje nahrávat obrázky a jiné soubory, na které se poté lze v textu odkazovat.

Přestože toho DokuWiki v základu nabízí mnoho, lze s ohledem na svou cílovou skupinu očekávat, že bude občas požadována nějaká další funkčnost. Nabízí tedy možnost dalšího rozšiřování pomocí tvorby vlastních zásuvných modulů.

2.1 Správce přístupových práv a správce médií

Stejně jako většina wiki systému je DokuWiki velmi otevřená. Kdokoliv může vytvářet, editovat či mazat stránky. Vzhledem k cílové skupině uživatelů je ale třeba někdy přístup na některé (nebo všechny) wiki stránky omezovat. K tomuto účelu slouží modul Access Control List (ACL).

Tento modul umožňuje jednoduše nastavovat různá oprávnění pro každou wiki stránku nebo jmenný prostor. Tato oprávnění se dají nastavovat jak pro celé skupiny, tak pro jednotlivé uživatele. Možných oprávnění je celkem 7: *none*, *read*, *edit*, *create*, *upload*, *delete* a *admin*. Každé vyšší oprávnění vždy obsahuje všechny nižší, přičemž práva *create*, *upload* a *delete* jsou nastavitelná pouze pro jmenné prostory.

DokuWiki také umožňuje pomocí správce médií nahrávání souborů do jednotlivých jmenných prostorů. Podle uživatelových práv je možné tyto soubory zobrazovat, mazat či přepisovat. Takto uložené soubory poté lze pomocí speciální wiki syntaxe vložit na libovolnou stránku. Obrázky, videa a zvukové soubory umí DokuWiki vykreslit přímo na

stránku. U ostatních typů souborů se zobrazí pouze odkaz na stáhnutí.

2.2 Tvorba rozšíření

Rozšíření [2] se tvoří ve stejném jazyce, v jakém je napsána DokuWiki, tedy v PHP. V současné době nabízí DokuWiki 7 různých typů zásuvných modulů:

- *Syntax* – umožňuje rozšiřovat základní syntaxi DokuWiki
- *Action* – díky tomuto typu modulu lze rozšiřovat vybranou interní funkcionalitu
- *Admin* – umožňuje přidat funkcionalitu do administrátorského menu
- *Helper* – funkce implementované v tomto typu lze používat z ostatních doplňků
- *Renderer* – pomocí něj lze nahradit standardní vykreslování do XHTML
- *Remote* – umožňuje přidávat vzdáleně volatelné funkce
- *Auth* – lze pomocí něj přidávat nové možnosti autentizace uživatelů (např. přes LDAP)

V jednom zásuvném modulu lze samozřejmě použít více různých typů. Název rozšíření udává jméno složky, ve které bude uloženo. Rozšíření (nebo jeho část) musí vždy být v souboru pojmenovaném podle požadovaného typu. Tento soubor by poté měl obsahovat třídu, která bude dědit od třídy vybraného typu a bude obsahovat vlastní funkčnost.

Složka se zásuvným modulem dále musí obsahovat soubor *plugin.info.txt*, ve kterém jsou podle přesně definované šablony zapsány rozšířené informace jako jméno autora, email nebo odkaz na webovou stránku o daném rozšíření. Jelikož DokuWiki podporuje překlad uživatelského rozhraní do jiných jazyků, může také rozšíření ve své složce obsahovat lokalizační soubory, které poté budou v zásuvném modulu automaticky zpřístupněny. V případě potřeby může také obsahovat soubory s JavaScriptem, CSS styly nebo konfigurační soubory, které umožňují integrovat nastavování vlastností rozšíření přímo ve správci nastavení DokuWiki.

Kromě metod, které vytvářené moduly vždy dědí od své nadřazené třídy, je jim k dispozici také několik globálních objektů a funkcí. Patří mezi ně například výše zmíněné metody pro přístup k lokalizačním souborům nebo metody pro zjištění různých informací o aktuálním prostředí (verze, jazyk, ...). Rozšíření mají také přístup ke globálním objektům, které obsahují informace o aktuální stránce či aktuálně přihlášeném uživateli. Nyní následuje detailnější popis typů modulů a funkcí, které budou v této práci použity.

2.2.1 Admin

Zásuvné moduly typu *Admin* umožňují přidávat funkcionalitu přístupnou přes administrační okno. Pomocí tohoto typu modulu se tedy nejčastější uživateli zpřístupňují různá nastavení pro ostatní části rozšíření. Modul se spustí kliknutím na odkaz v již zmíněném administrátorském menu. Tím se zavolají následující funkce, které musí být povinně implementovány:

- *handle()* – v této metodě by se měli zpracovávat požadavky od uživatelů, což zahrnuje například odeslání nějakého formuláře
- *html()* – pomocí této metody se potom tvoří obsah výsledné stránky (rozhraní mezi uživatelem a systémem)

2.2.2 Syntax

Pomocí zásuvných modulů typu *Syntax* lze rozšiřovat základní syntaxi DokuWiki. Ve své podstatě umožňují nahradit nějaký textový řetězec definovaný pomocí regulárního výrazu nějakým vlastním. Kromě rozšiřování možností formátování textu toto také umožňuje vkládání jakéhokoliv obsahu na wiki stránku pomocí nějakého speciálního řetězce. Tohoto se dá využít například pro vkládání videí nebo komplexních formulářů. Moduly typu *Syntax* musí mimo jiné implementovat hlavně tyto tři funkce:

- *connectTo(...)* – pomocí této funkci se definuje výše zmíněný regulární výraz
- *handle(...)* – tato metoda se volá, pokud se na wiki stránce najde nějaký obsah, který odpovídá definovanému regulárnímu výrazu
- *render(...)* – pomocí této metody se potom tvoří výsledný obsah, kterým se nahradí nalezený text

Je také důležité zmínit, že výsledek metody *handle()* se ukládá do vyrovnávací paměti, a poté je předáván metodě *render()*. Většinu zpracovávání je tedy dobré provádět v první zmiňované funkci.

Metodě *render()* je také předáván objekt typu *Doku_Renderer*, který vykresluje zrovna zobrazovanou stránku. Právě pomocí tohoto objektu lze na stránku přidávat další vlastní obsah. Pokud navíc potřebujeme na stránku vložit nějaký text, který obsahuje wiki syntaxi, tento objekt nám umožní jednoduše vykreslit (tedy zkonvertovat danou wiki syntaxi na její HTML reprezentaci).

2.2.3 Action

Zásuvné moduly typu *Action* umožňují rozšiřovat různé interní části DokuWiki v průběhu zpracovávání stránek. Tyto moduly jsou načteny před každým procesem (například načtení stránky). Každý takový modul si může registrovat typy událostí, které chce obsluhovat. Když je událost signalizována, volají se postupně všechny zásuvné moduly, které na tuto událost chtějí reagovat. Existují přitom dvě možnosti, kdy se doplněk může zavolat – buď před danou událostí, nebo až po tom, co se událost vykoná. Doplněk musí vždy implementovat metodu *register()*, ve které se provede registrace požadovaných událostí a poté pro každou registrovanou událost jednu funkci, která se bude při dané události volat.

Dostupných událostí, na které lze reagovat je značné množství. Jsou rozděleny podle typu do několika kategorií. Lze mezi nimi nalézt například události, které umožňují modifikovat standardní formuláře pro přihlašování, editaci stránky či nahrávání souborů nebo třeba události, které umožňují měnit chování správce přístupových práv a správce médií. Zaregistrované metodě pro obsluhu jsou poté vždy předávány dodatečné informace o dané události a případně i proměnné, pomocí kterých lze ovlivňovat její výsledek. Pro naše potřeby bude třeba obsluhovat tři události: *AJAX_CALL_UNKNOWN*, *AUTH_ACL_CHECK* a *FETCH_MEDIA_STATUS*.

První zmiňovaná slouží pro odchytávání AJAX požadavků. [4] Požadavek musí vždy obsahovat parametr, který určuje jeho název a musí se posílat na speciální adresu v DokuWiki, kde se zpracuje a vyvolá tuto událost. Metoda pro obsluhu poté obdrží název požadavku spolu s dalšími případnými parametry a má možnost poslat zpátky odpověď.

Druhá událost umožňuje měnit chování přístupových práv. Je vyvolávána pokaždé, když se v průběhu vykreslování nějaké stránky nebo zpracovávání nějakého požadavku zjišťuje,

jaká práva má aktuálně přihlášený uživatel v daném kontextu. Metodě pro obsluhu této události jsou poté předávány informace o tomto uživateli a kontextu, a má možnost změnit jaký typ práv se navrátí.

Poslední událost poté umožňuje upravovat chování DokuWiki při pokusu o stahování souborů.

2.2.4 JavaScript, CSS

V případě, že chceme v rozšíření využít JavaScriptu nebo CSS, je třeba tyto soubory správně umístit a pojmenovat. V obou případech se umísťují přímo do složky s rozšířením. V případě JavaScriptu se tento soubor musí jmenovat *script.js*, v případě CSS je to *style.css*.

DokuWiki všechny soubory s JavaScriptem a CSS (jak interní tak ze všech rozšíření) vždy pospojuje do jednoho souboru a na každé zobrazované stránce jsou poté dostupné všechny styly či funkce zároveň. Je tedy nutné dbát na to, abychom při tvorbě těchto souborů volili unikátní identifikátory, u kterých nedojde ke konfliktu s těmi již existujícími v interních částech DokuWiki nebo v ostatních modulech.

2.2.5 Lokalizace

Jak bylo zmíněno na začátku této kapitoly, uživatelské rozhraní zásuvných modulů pro DokuWiki lze překládat. Soubory pro lokalizaci musí být uloženy v adresáři pojmenovaném podle daného jazyka (například */lang/cs/* pro češtinu). V souboru *lang.php* může být uložen slovník, ke kterému se poté v rozšíření přistupuje pomocí zděděné metody *getLang(klíč)*, která sama podle aktuálně nastaveného jazyka vrátí hodnotu podle zadaného klíče ze správného adresáře s lokalizací. Stejná metoda je v případě potřeby dostupná i v JavaScriptu.

2.2.6 Globální proměnné a funkce

DokuWiki také umožňuje přistupovat k několika užitečným globálním proměnným, které umožňují získávat dodatečné informace o aktuálním kontextu. Z těch zajímavých pro tuto práci jsou to:

- *$INPUT* – nabízí typově bezpečný přístup k proměnným z POST a GET požadavků s možností specifikace návratové hodnoty v případě, že hledaná proměnná neexistuje
- *$INFO* – zde jsou uloženy veškeré informace o aktuální stránce
- *$USERINFO* – obsahuje informace o aktuálně přihlášeném uživateli

Je také možné využívat velké množství globálních funkcí, kterými lze ovládat jednotlivé části DokuWiki. Jedná se například o funkce pro kontrolu přístupových práv, nahrávání souborů do správce médií či přidávání nebo mazání jmenných prostorů a jednotlivých stránek.

Kapitola 3

Návrh zásuvného modulu

V této kapitole se budeme zabývat popisem všeho, co by tento modul měl umět. Jelikož se bude jednat o katalog zdrojů, bylo by vhodné uvést, co v kontextu této práce rozumíme pod pojmem zdroj. Jako zdroj označujeme nějaký ucelený soubor informací či myšlenek, na který se lze odkazovat. Jedná se tedy převážně o textové dokumenty jako knihy, sborníky či časopisy, ale dají se tam počítat i audiovizuální zdroje jako například filmy.

Dále by bylo vhodné uvést specifikaci požadavků na výsledné rozšíření. Tedy základní přehled toho, co by tento zásuvný modul měl umět. Nejzákladnější funkcí by měla být základní správa zdrojů v databázi. Tedy možnost vyhledávání, jejich editace, přidávání, mazání a zobrazování informací o nich. Přidávání nových zdrojů by přitom mělo být částečně automatické – mělo by být podporováno získávání informací z existujících citací (například ve formátu BibTeX).

U každého zdroje by měla být možnost nahrát k němu samotný dokument zdroje nebo přidat odkaz na jeho stažení, přičemž musí existovat možnost nastavení data, od kterého bude tento odkaz či dokument přístupný. Tyto dokumenty smí být do daného data přístupné pouze autorizovaným uživatelům a ne veřejnosti. Důvodem jsou časté licenční omezení těchto dokumentů, které zveřejňování povolují až po určité době. Toto byla zároveň hlavní motivace pro vznik tohoto zásuvného modulu. Při zobrazení informací o zdroji by mělo být možné pomocí uložených informací vytvořit citaci (minimálně ve formátu BibTeX). Také by zde měla být možnost k danému zdroji přidávat komentáře, ve kterých by mělo být možné využívat dostupné wiki syntaxe, především kvůli případné podpoře vykreslování matematických vzorců pomocí jiného zásuvného modulu.

Modul musí podporovat možnost spravovat více databází zdrojů, přičemž k jejich ukládání by neměl být použit databázový server, ale měly by se ukládat přímo do souboru. Tyto databáze by také měli jít importovat či exportovat z/do různých tabulkových formátů jako jsou XLS nebo CSV.

Posledním požadavkem je, aby přístup ke všem vlastnostem tohoto modulu správně respektoval nastavení přístupových práv v administrátorském menu. Celý zásuvný modul bude rozdělen do několika částí, které jsou popsány v následujících kapitolách.

3.1 Administrátorská část

Základním stavebním prvkem tohoto modulu bude stránka v administrační části DokuWiki – bude se tedy jednat o rozšíření typu *Admin* popsané v kapitole 2.2.1. Tato stránka bude obsahovat základní nastavení a konfigurace tohoto rozšíření.

Vzhledem k tomu, že modul bude muset umět pracovat s více databázemi zdrojů, je nutné, aby je v této části bylo možné spravovat. To zahrnuje především jejich vytváření a mazání. Při vytváření bude možné vždy zvolit jméno databáze, podle kterého se poté bude vkládat na wiki stránku. Dále je také bude nutné každé vytvářené databázi přiřadit jmenný prostor, do kterého se budou ukládat případné nahrané soubory a kde také bude uložen samotný soubor s touto databází.

V této části modulu také bude umožněno do zvolené databáze hromadně importovat nějaká již existující data. Bude se jednat alespoň o jeden z formátu, které využívají tabulkové procesory (tedy XLS, XLSX, ODS nebo CSV). Import bude probíhat jednoduchým způsobem. První řádek importované tabulky bude muset obsahovat hlavičky pojmenované podle přesně daného vzoru podle toho, co za informace se v daném sloupci nachází. V každém řádku budou poté muset být informace o jednom zdroji. Modul takovýto soubor bude poté řádek po řádku číst a daná data importovat do správných míst v databázi. Do stejných formátů bude možné existující databáze z tohoto modulu i exportovat. Ať už z důvodu zálohy či využití v jiných aplikacích.

Kromě správy databází zde také bude možné nastavit část textového řetězce, pomocí kterého se budou databáze vkládat na vlastní wiki stránky. V rámci zachování konzistence s ostatními doplňky bude výsledný řetězec ve formátu {{*nastavitelnýŘetězec:názevDatabáze*}}. Neměnný přednastavený řetězec nebude zvolen z prostého důvod – zamezení případné nekompatibility s ostatními rozšířeními, které by mohlo nešťastnou náhodou chtít využít stejný formát pro vkládání svého obsahu.

3.2 Vyhledávání mezi zdroji

Hlavní části tohoto rozšíření bude formulář, pomocí kterého bude možné vyhledávat ve zdrojích v určité databázi. Tento formulář se vloží na jakoukoliv wiki stránku pomocí textového řetězce popsaného výše, čehož se docílí díky využití zásuvného modulu typu *Syntax*, který byl popsán v kapitole 2.2.2. Pokud se na nějaké stránce objeví daný nastavitelný řetězec, tak se analyzuje (tedy získá se z něj jméno databáze) a nahradí vlastním formulářem. Vložený formulář bude umožňovat vyhledávání podle všech dostupných kritérií. Výsledný seznam vyhledaných zdrojů se zobrazí jako přehledná tabulka se základními informacemi o každém zdroji, která se bude dát řadit podle zobrazených sloupců. Součástí každého řádku s informacemi o zdroji bude odkaz na stránku s detaily.

Součástí tohoto formuláře bude zároveň muset být odkaz na přidání nového záznamu do databáze. Stránka pro přidání nového zdroje bude obsahovat výběr typu zdroje, který chceme přidat, a podle kterého se nám poté budou nabízet k vyplnění informace, které lze k danému typu přiřadit. Dále také bude možnost tento typ ignorovat a zobrazit pole všechna, pro speciální případy zdrojů, které se nedají zařadit do žádného z dostupných. Bude zde též možnost vložit existující citaci ve formátu BibTeX, která bude zanalyzována a automaticky formulář předvyplní. Po úspěšném přidání zdroje bude následovat automatické přesměrování na stránku s detaily o tomto zdroji, která je popsána v další kapitole.

3.3 Zobrazení informací o zdroji

Po úspěšném vyhledání nebo přidání požadovaného zdroje se dostáváme k další části tohoto modulu. Bude se jednat o stránku, která bude zobrazovat informace o jednom samostatném zdroji. Její hlavní funkcí je zobrazení všech dostupných informací. Zároveň zde také

bude možnost, pokud stránku navštíví uživatel s dostatečnými oprávněními, tyto informace editovat nebo případně celý zdroj z databáze smazat.

Kromě zobrazení informací zde také bude možnost exportování citací. Na základě informací o tomto zdroji uložených v databázi se vždy automaticky vygeneruje citace ve formátu podle české normy ISO 690, pro případné rychlé vkládání do různých kratších prací či příspěvků. Dále zde pak půjdou informace o zdroji exportovat do formátu BibTeX, který umožní daný zdroj jednoduše a rychle citovat v dokumentech psaných v LaTeXu. Oba typy vygenerovaných citací bude možné jednoduše pomocí tlačítka zkopírovat do schránky.

Součástí této stránky také bude možnost spravovat soubory přiřazené k aktuálně zobrazovanému zdroji. Všechny takové soubory zde budou vypsány spolu s odkazem na jejich stáhnutí. Bude zde také možnost další dokumenty přidávat přímo do úložiště DokuWiki, přesněji do jmenného prostoru, který byl přiřazen zobrazované databázi. Zároveň bude možné každému souboru přiřadit datum, od kterého bude možné zobrazovat odkaz na něj veřejnosti. Kromě přidávání zde také samozřejmě bude možnost daný dokument smazat. Souborů bude možné ke každému zdroji přidávat libovolné množství.

Poslední funkcí této části modulu bude podpora pro komentování zobrazovaného zdroje. Pod informacemi o zdroji bude dostupný formulář, který umožní přidávat vlastní příspěvky. Každý komentář bude kromě vlastního textu vždy obsahovat datum jeho přidání a také jméno autora, který příspěvek odeslal. Přidané komentáře bude možné autorem a případně administrátory či moderátory jednoduše editovat a mazat. Text jednotlivých příspěvků se bude vykreslovat stejným způsobem jako normální wiki stránky, tedy tak, aby zde bylo možné využívat wiki syntaxi. To je nutné hlavně z důvodu toto, že se bude jednat převážně o databáze odborných technických zdrojů, takže zde budou uživatelé občas chtít zobrazit například nějaké matematické výrazy. Bude tedy poté stačit nainstalovat nějaký již existující zásuvný modul, který pomocí wiki syntaxe umožňuje takové matematické výrazy vykreslovat.

3.4 Přístupová práva

Výsledný zásuvný modul musí správně respektovat nastavení přístupových práv. Možnost zobrazovat či editovat záznamy v databázích tedy musí mít jen vybraní uživatelé. Stejně tak bude třeba kontrolovat přístup k souborům ve správci médií.

Informace o přístupových právech pro konkrétní stránku, jmenný prostor nebo soubor lze zjistit velmi jednoduše – zavoláním globální metody *auth_quickaclcheck(id)*, který vždy vrací konstantu určující úroveň práv pro aktuálního uživatele v daném kontextu. Kontrola práv pomocí této metody a následné zobrazení či nezobrazení obsahu by nám v normálním případě stačilo. Jelikož ale potřebujeme zajistit, aby se soubory přidané k dokumentům zobrazovali ve správci médií, pouze pokud již uběhla nastavená lhůta, bude potřeba chování této metody poupravit, aby vracela správná práva pro naše soubory, i pokud se bude volat z ostatních části DokuWiki, které normálně nemůžeme kontrolovat. Toho docílíme zachytáváním speciální události, která se vyvolává po zavolání této metody. Použijeme k tomu zásuvný modul typu *Action*, který byl popsán v kapitole 2.2.3.

Každá databáze bude svázána s nastavitelným jmenným prostorem, do kterého se budou ukládat jak soubory, tak samotný soubor s databází. Práva pro přístup k této databázi a jejím souborům tedy budou určený nastavením ve správci přístupových práv právě pro přiřazený jmenný prostor. Budeme uvažovat pouze čtyři základní úrovně práv, podle kterých se bude zpřístupňovat určitá funkčnost tohoto zásuvného modulu:

- *none* – bez přístupu
- *read* – vyhledávání, zobrazení informací o zdrojích včetně komentářů, exportování citací, zobrazování veřejně přístupných souborů
- *edit* – úprava zdrojů, přidávání nový zdrojů, zobrazování všech souborů (i těch, které by se podle nastaveného data ještě neměly zobrazovat), přidávání komentářů, mazání a editace vlastních komentářů
- *admin* – správa databází, mazání a editace všech komentářů

3.5 Způsob uložení dat

Jedním z problémů, který je třeba vyřešit, je jakým způsobem se budou databáze ukládat. Nabízí se několik možností. První z nich je použití klasického databázového systému. V případě PHP se jedná hlavně o systém MySQL. Výhodou tohoto řešení je relativní jednoduchost jeho použití – PHP již v základu obsahuje knihovny a funkce pro práci s těmito systémy. Vyhledávání, editace či vkládání nových záznamů probíhá pomocí jednoduchých SQL dotazů. Jelikož je ale jednou z hlavních vlastností DokuWiki to, že nepotřebuje ke svému běhu žádný speciální databázový server, tak tato možnost nebude v našem případě úplně nejvhodnější.

Pokud zavrhneme použití jakéhokoliv databázového serveru, zbývá nám pouze jedna možnost – ukládání přímo do souboru v nějakém formátu. Jednou cestou by bylo využití nějakého standardního formátu pro serializaci dat, jako jsou například formáty XML nebo JSON. Tyto formáty jsou velmi oblíbené a často používané, což je jedna z jejich hlavních výhod. Funkce pro práci s nimi jsou totiž dostupné v podstatě v každém programovacím jazyce, včetně PHP. Zároveň jsou takto zformátovaná data čitelná a upravitelná i v běžném textovém editoru. Nevýhodou oproti databázím je nutnost implementace vlastních algoritmů editaci či pro vyhledávání podle více kritérií, přičemž při každé operaci je nutné daný soubor otevřít a celý projít. Neexistuje zde žádná automatická indexace.

Poslední možností je využití databázové knihovny SQLite. Jedná se o relační databázový systém obsažený v relativně malé knihovně. Klasické databázové systémy, které fungují na principu klient-server, ke svému používání vždy vyžadují speciální samostatný proces – databázový server, pomocí kterého se s databází pracuje, a který zajišťuje ukládání dat. V případě SQLite se ale databáze ukládají přímo na disk do souborů. SQL dotazy se nad nimi poté provádějí pomocí knihovních funkcí. Pro použití tedy není vyžadováno nic jiného než tato knihovna. Vzhledem k podpoře indexace je tato cesta ukládání dat také pravděpodobně efektivnější. Toto řešení tedy spojuje výhody obou výše popsaných – jednoduché a efektivní používání (vyhledávání, vkládání, editace, ...) díky podpoře SQL dotazů a ukládání přímo do souboru (což umožňuje třeba jednodušší zálohování). Nevýhodou je, že tato knihovna nemusí být dostupná všude. PHP sice od jisté verze obsahuje podporu pro tuto knihovnu, ale ne na všech strojích musí být povolená.

Ideálním řešením bude tedy vytvořit rozhraní pro jednotnou práci s databází, aby bylo možné jednoduše přidávat podporu pro jednotlivé formáty. Implementovány tedy budou nějaká z výše uvedených řešení a uživateli bude při vytváření nové databáze vždy dána možnost výběru formátu, v jakém se bude ukládat.

3.6 Bibliografické údaje

Po vyřešení způsobu jakým se budou data ukládat, se nabízí další otázka – jaká data vlastně budeme muset uchovávat. Existuje totiž několik různých druhů zdrojů a u každého je možné uvádět jiné druhy povinných a nepovinných údajů. Při přidávání nových zdrojů do databáze tedy bude nutné dát uživateli na výběr z určitých typů zdroje a podle toho dále nabídnout k vyplnění všechny informace, které pro daný typ dávají smysl. Je tedy třeba ustanovit si seznam typů zdrojů, které bude možné do databáze uložit a ke každému také seznam informací, které k němu bude možné přiřadit.

Jako směrodatnou budeme brát českou citační normu ČSN ISO 690. Tato norma z roku 2011 definuje formát, v jakém se mají jednotlivé typy zdrojů citovat. Uvádí vždy seznam povinných a nepovinných údajů spolu s jeho správným pořadím. Je tedy třeba vybrat takový soubor typů údajů, aby se do nich daly zapsat všechny potřebné informace u všech typů zdrojů. Typů zdrojů je definováno značné množství a potřebné údaje se ve většině případů překrývají. Některé z typů můžeme pro naše potřeby vypustit – například film, seriál nebo grafické dílo. Ve výsledku se tedy bude jednat o tyto hlavní typy zdrojů:

- kniha a její části či kapitoly
- sborníky z konferencí a jednotlivé články v nich
- periodika a jejich články
- webové stránky
- kvalifikační práce – bakalářské a diplomové práce

Z těchto typů zdrojů je tedy potřeba vybrat výsledné typy údajů. Jak je psáno výše, drtivá většina údajů se překrývá. Konkrétně u kvalifikační práce je několik unikátních údajů, pro které by ale bylo možno využít ostatní podobná pole, které naopak kvalifikační práce nevyžaduje.

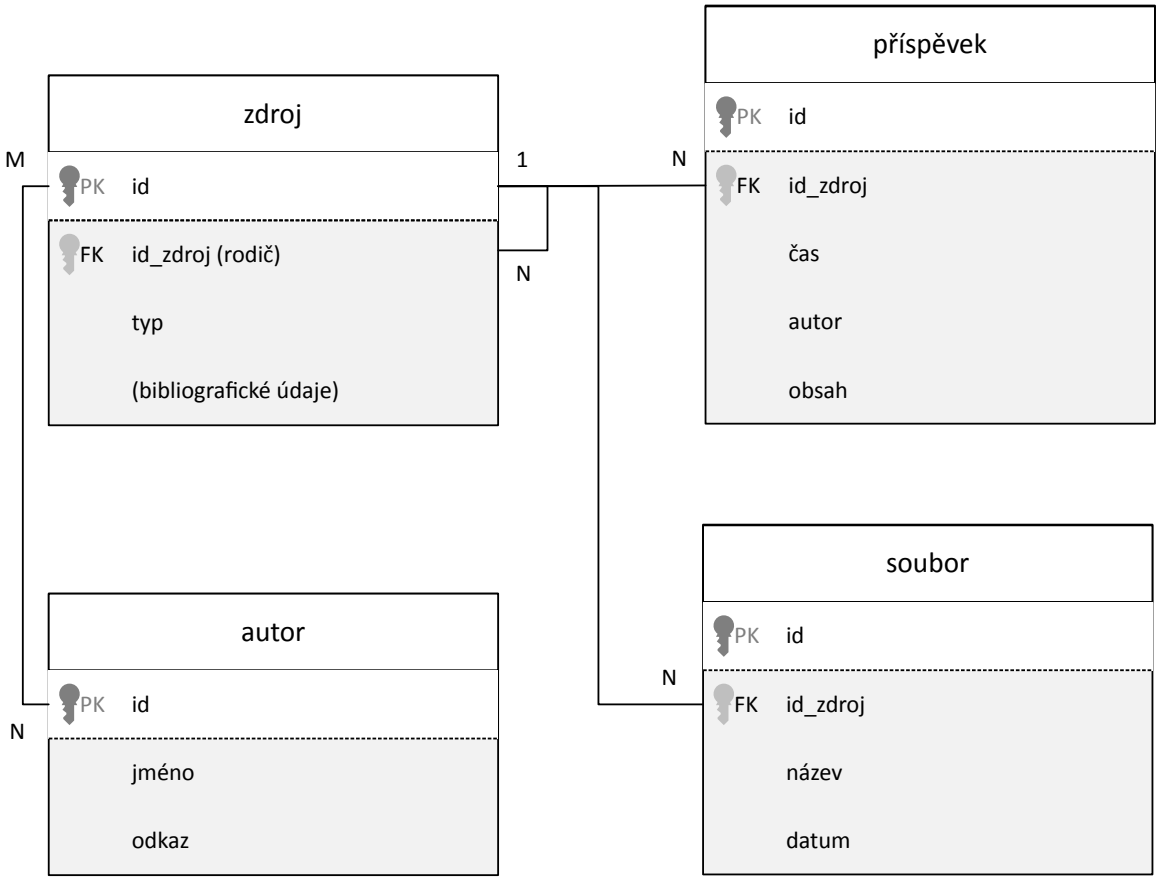
Prvním povinným údajem u každého zdroje jsou jeho autoři. Těch ale může být uvedeno větší množství než jen jeden. S touto položkou tedy bude třeba pracovat odlišným způsobem než se všemi ostatními, protože ji bude třeba ukládat jako seznam. U kvalifikační práce, je navíc třeba ještě ukládat jméno vedoucího práce. Další potřebné údaje již jen rámcově:

- titulek, případně titulek nadřazeného zdroje (například pro kapitolu v knize)
- vydání (u kvalifikační práce případně její druh)
- místo publikování (případně místo odevzdání u kvalifikační práce)
- nakladatel (nebo název institutu u kvalifikační práce)
- datum publikování (a datum odevzdání u kvalifikační práce)
- edice a číslování
- číslo kapitoly a rozsah stránek
- identifikátory – ISBN pro knihy, ISSN pro periodika, DOI pro elektronické dokumenty
- dostupnost – například URL u webové stránky
- poznámky

3.7 Návrh datového modulu

Z údajů uvedených v předchozí kapitole tedy již můžeme sestavit návrh datového modelu, pomocí kterého by se poté data ukládala – ať už do XML či JSON souborů, nebo do databází pomocí knihoven SQLite.

Datový model bude obsahovat čtyři entity. První bude reprezentovat samotný zdroj, druhá bude reprezentovat komentář, třetí bude reprezentovat soubor a poslední bude reprezentovat jednoho autora. U entity komentáře bude třeba uchovávat jeho čas vytvoření, jméno autora, který příspěvek napsal, a poté samotný text. Entita souboru bude muset obsahovat informace o jeho názvu a o datu, od které může být zobrazován veřejnosti. U entity autora bude jeho jméno a případný odkaz na jeho stránky. Entita zdroje poté bude obsahovat všechny bibliografické údaje vypsané v předchozí kapitole spolu s typem zdroje. Může také obsahovat odkaz na nadřazený zdroj – například v případě kapitoly v knize odkaz na samotnou knihu, ve které se kapitola nachází. Výsledný model je vidět na obrázku 3.1.



Obrázek 3.1: Datový model

Při ukládání dat pomocí knihoven SQLite by bylo možné vytvořit tabulky v databázi přesně podle tohoto modelu. Pro implementaci ukládání do formátu XML či JSON bude udělána malá změna. Entity souboru a komentáře nebudou muset obsahovat cizí klíče, protože budou strukturou dokumentu vždy svázány s určitým zdrojem. Místo identifikátoru poté bude možné využít u souboru jeho název a u komentáře jeho datum vytvoření. Schéma výsledného XML souboru je na obrázku 3.2 (pro jednoduchost je zde uvedena pouze jeho část).

```
<xs:element name="zdroj">
    <xs:complexType>
        <xs:all minOccurs="0" maxOccurs="unbounded">
            <xs:element name="id" type="xs:integer"/>
            <xs:element name="type" type="xs:string"/>
            <xs:element name="author" type="xs:integer"/>
            <xs:element name="comment">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="author" type="xs:string"/>
                        <xs:element name="date" type="xs:date"/>
                        <xs:element name="text" type="xs:string"/>
                    </xs:all>
                </xs:complexType>
            </xs:element>
            <xs:element name="file">
                <xs:complexType>
                    <xs:all>
                        <xs:element name="name" type="xs:string"/>
                        <xs:element name="date" type="xs:date"/>
                    </xs:all>
                </xs:complexType>
            </xs:element>
        </xs:choice>
    </xs:complexType>
</xs:element>
<xs:element name="author">
    <xs:complexType>
        <xs:all>
            <xs:element name="name" type="xs:string"/>
            <xs:element name="www" type="xs:string"/>
        </xs:all>
    </xs:complexType>
</xs:element>
```

Obrázek 3.2: XML Schéma

Kapitola 4

Implementace

V této kapitole bude detailněji popsána samotná implementace zásuvného modulu navrženého v předchozích kapitolách.

4.1 Uložení nastavení

Kromě samotných databází je třeba ukládat dvě zásadní věci – řetězec, pomocí kterého se budou databáze vkládat na wiki stránky (viz kapitola 3.1) a seznam existujících databází spolu se jmennými prostory, které se k nim při vytváření přiřadí. Bylo tedy třeba implementovat nějaký jednoduchý způsob, jak těchto několik hodnot typu klíč-hodnota ukládat do souboru.

Za tímto účelem byla vytvořena třída *Config*, která dané informace umožňuje ukládat do souboru ve formátu JSON. Ten byl zvolen hlavně pro svoji jednoduchost používání. Tato třída obsahuje dvě statické metody:

- *get(klíč)* – pomocí metody *json_decode(...)* načte soubor s konfigurací a poté vrátí hodnotu pro klíč předaný v parametru
- *set(klíč, hodnota)* – opět načte soubor s nastavením, nastaví požadovanou hodnotu klíče a pomocí metody *json_encode(...)* uloží data zpátky do souboru

Pokud tyto metody nenaleznou konfigurační soubor (*config.json*), například při nové instalaci, vytvoří nový, který bude obsahovat implicitní nastavení – tedy jednu položku pro řetězec na vkládání databází.

4.2 Práce s databázemi

Jak bylo popsáno v kapitole 3.5, pro práci s databázemi bylo vytvořeno rozhraní, které umožňuje přidávat podporu více formátů. S databázemi se poté může pracovat jednotným způsobem, bez ohledu na zvolený formát. Implementovány byly dvě možnosti uložení dat - ve značkovacím jazyce XML a ve formátu JSON.

Databases

Základním přístupovým bodem pro práci s databázemi je statická třída Databases. V této třídě je definováno asociativní pole, které přiřazuje dostupné implementace formátů da-

tabází ke koncovkám souborů, které vytváří. Dále obsahuje metody pro základní správu databází:

- *all()* – vrací seznam všech existujících databází spolu s jejími jmennými prostory
- *set(název, formát, jmennýProstor)* – tato metoda podle pole popsaného výše vybere správnou implementaci databáze podle předaného formátu, vytvoří v DokuWiki potřebný jmenný prostor, uloží do něj novou databázi a přidá pro ni položku do konfiguračního souboru
- *get(název)* – podle konfiguračního souboru zjistí, v jakém jmenném prostoru se hledaná databáze nachází, a podle koncovky nalezeného souboru vrací správnou implementaci databáze
- *remove(název)* – opět vyhledá jmenný prostor a soubor s databází v něm, tento soubor smaže, patřičně upraví konfigurační soubor a nakonec smaže i celý jmenný prostor, ve kterém se databáze nacházela, pokud je jinak prázdný

Typ databáze je tedy určen koncovkou souboru a podle toho se vždy vybírá správná třída, která s tímto souborem bude pracovat. Jednotlivé implementace jsou popsány v následujících odstavcích.

Database_Base

Tato abstraktní třída slouží jako bázová pro všechny implementace. Deklaruje několik metod, které musí být v odvozených třídách vždy implementovány. Pomocí tohoto rozhraní se poté s databázemi jednotně pracuje bez ohledu na jejich formát. Mimo toto rozhraní uchovává dvě hodnoty, které se jí předají v konstruktoru – její název a jmenný prostor. Obsahuje také metodu *getPath()*, která pomocí právě těchto dvou hodnot navrací celou cestu k souboru, ve kterém je databáze uložená. Rozhraní, které je nutné implementovat, se poté skládá z těchto metod:

- *get(id)* – vyhledá zdroj podle jeho *id* a navrací asociativní pole obsahující všechny jeho dostupné informace včetně seznamu autorů, komentářů a souborů
- *find(data)* – vyhledá všechny zdroje, které odpovídají parametrům předaným v asociativním poli *data* a navrátí jejich seznam
- *add(data)* – přidá do databáze novou položku s informacemi podle parametru *data* a navrací její *id*
- *edit(id, data)* – vyhledá požadovaný zdroj a změní v něm předané vlastnosti
- *remove(id)* – smaže daný zdroj
- *add/edit/removeComment(...)* – metody pro přidávání, editaci a mazání komentářů
- *add/edit/removeFile(...)* – analogicky metody pro práci se soubory
- *isPublicFile(name)* – kontroluje, jestli daný soubor v databázi existuje a jestli je možné jej zobrazovat veřejnosti

Pro přidání nového způsobu ukládání databází je tedy třeba implementovat třídu, které bude dědit z třídy *Database_Base*, a přiřadit jí koncovku v seznamu třídy *Databases*.

Database_Xml

Prvním formátem databáze, který byl implementován, je značkovací jazyk XML. Pro práci s tímto formátem byla využita vestavěná PHP knihovna SimpleXML. Ta umožňuje jednoduchým způsobem přidávat, měnit a číst jednotlivé elementy či atributy v XML souboru. V konstruktoru je tedy vytvořen objekt typu *SimpleXMLElement*, do kterého je předána cesta k souboru s databází, a pomocí něj poté metody implementující výše popsané rozhraní s databází pracují.

SimpleXML zároveň nabízí možnost použití standardu XPath. Tento jazyk umožňuje vyhledávat v XML souborech podle speciálních dotazů a navrací kolekci výsledků, které takovému dotazu odpovídají. Toho bylo využito pro jednoduché vyhledávání zdrojů podle id a pro vyhledávání mezi komentáři a soubory. Pro vyhledávání podle více parametrů bylo nutné implementovat vlastní algoritmus, aby bylo například možné při porovnávání záznamu s hledanou hodnotou ignorovat velikost písmen.

Database_Json

Druhým implementovaným formátem je JSON. Hlavní výhoda tohoto formátu je jeho značná jednoduchost a čitelnost výsledných souborů. Pro práci s tímto formátem byly využity vestavěné funkce PHP, který byly zmíněny v kapitole 4.1.

V konstruktoru je vždy celý soubor s databází načten a následně dekódován pomocí metody *json_decode()*. Výsledkem je struktura asociativních polí, které lze běžným způsobem procházet, měnit v nich informace nebo je mazat. V případě upravování je po jejich dokončení tato struktura pomocí metody *json_encode()* zakódována a uložena zpátky na disk.

Konkurentní přístup

S databázemi může samozřejmě pracovat více lidí současně. Každá operace prováděná nad jakoukoliv databází nějakou dobu trvá. Implementované metody navíc vždy vyžadují při čtení načítat celý soubor a při jakékoliv změně opět zapisovat celý soubor. Pokud by se dva uživatelé snažili například měnit nějaké informace ve stejný okamžik, tak by se mohlo stát, že se průběhu zapisování souboru na disk jednou operací bude zároveň snažit o zápis i operace druhá, což by pravděpodobně způsobovalo chyby, jako například jen částečně zapsaný soubor, nebo spojení dvou výstupů v jeden.

Bylo tedy třeba takovémuto chování zamezit. Za tímto účelem byla využita metoda pro uzamykání souborů *flock()*, která je běžnou součástí nabídky funkcí PHP. Funkci lze předávat parametr, který určuje, jestli se jedná o operaci čtení nebo zápisu. Před každým čtením nebo zápisem je tedy tato metoda volána a po ukončení opera je soubor odemčen. V případě, že se snaží nějaká operace databázový soubor zapisovat, volání této funkce automaticky počká, dokud nejsou dokončeny všechna čtení. Analogicky pokud se operace snaží soubor číst, tato metoda počká, dokud ostatní operace nedokončí zápis.

4.3 Zásuvný modul typu Admin

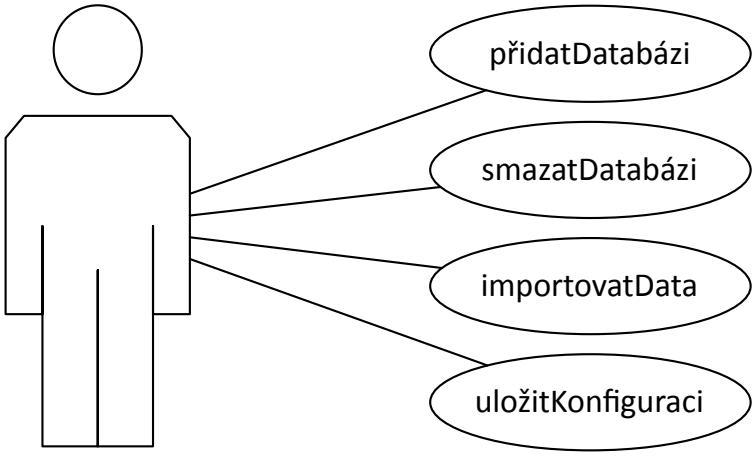
Spravování databází a nastavení zajišťuje třída *admin_plugin_publications*, která je potomkem třídy *DokuWiki_Admin_Plugin*. Ta přidává položku do administrátorského menu a byla popsána v kapitole 2.2.1. Obsahuje dvě hlavní metody – *html()* a *handle()*.

html()

V této metodě je generován veškerý obsah administrační stránky. Tedy formulář pro nastavování vkládacího řetězce a pro správu databází. HTML kód pro tento formulář je jednoduše tisknut na standardní výstup. Formulář obsahuje pole pro vkládací řetězec, seznam databází, tlačítka pro jejich mazání, přidávání (spolu s poli pro název databáze a jmenný prostor) a pro hromadný import z vybraného souboru.

handle()

Tato metoda řeší situaci po odeslání výše zmíněného formuláře. Podle typu stisknutého tlačítka, který se zjistí z parametru v URL, se provede odpovídající akce. Jejich přehled je vidět v diagramu na obrázku 4.1.



Obrázek 4.1: Diagram použití Admin

4.3.1 Hromadný import

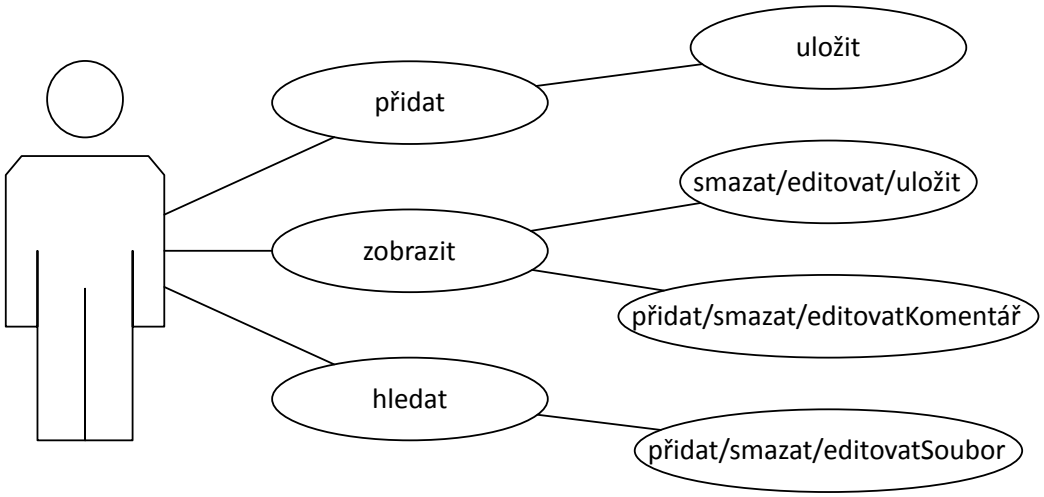
Pro hromadné importování existujících záznamů do zvolené databáze byla použita open source knihovna PHPExcel. [5] Jedná se o jednu z knihoven kolekce PHPOffice, která umožňuje pracovat s tabulkovými formáty v jazyce PHP. Podporuje všechny běžné formáty – tedy XLS, XLSX, ODS nebo CSV. Nabízí rozhraní jak pro čtení, tak pro zápis, přičemž podporuje například i procházení více listy a editaci formátování.

Samotné importování je řešeno ve třídě *Importing*. Po vybrání databáze a nahrání souboru s daty k importaci se zavolá metoda *import(...)* z této třídy, které se předá cesta k souboru a název vybrané databáze. Metoda pro otvírání souborů knihovny PHPExcel automaticky rozpozná jeho formát a navrací správnou implementaci rozhraní pro práci s daným formátem.

V otevřené tabulce se poté přečte obsah prvního řádku, který musí obsahovat hlavičky pro dané sloupce. Pokud nějaká hlavička nesouhlasí s žádným s předepsaných vzorů, sloupec se ignoruje. Následně se jednoduše projde zbytek tabulky řádek po řádku a podle hlaviček nalezených v předchozím kroku se generují záznamy do databáze s patřičnými informacemi.

4.4 Zásuvný modul typu Syntax

Vyhledávání, zobrazování a přidávání zdrojů zajišťuje třída *syntax_plugin_publications*, která je potomkem třídy *DokuWiki_Syntax_Plugin* popsané v kapitole 2.2.2. Tato třída si v metodě *connectTo()* registruje regulární výraz obsahující vkládací řetězec z konfiguračního souboru. Když DokuWiki nalezne na wiki stránce řetězec odpovídající danému regulárnímu výrazu, zavolá funkci *handle()* v tomto modulu, která z nalezeného řetězce extrahuje název databáze. Metodě *render()* je poté nalezený název předán spolu s objektem, který zajišťuje vykreslování aktuální stránky. Do tohoto objektu se poté generuje patřičný obsah, kterým se původní řetězec nahradí. Podle parametru v URL se bude vždy vykreslovat jeden ze tří módů zobrazení. Jejich přehled spolu s dostupnými akcemi je vidět v diagramu na obrázku 4.2.



Obrázek 4.2: Diagram použití Syntax

4.4.1 Vyhledávání

Implicitním módem, který se zobrazí, je vyhledávání mezi zdroji. Obsahuje formulář s několika poli, pomocí kterých se dá vyhledávat. Kromě tlačítka pro samotné vyhledávání je zde také ještě možnost přejití na mód pro přidávání nových zdrojů. Pod formulářem se poté generuje tabulka s nalezenými zdroji. Pokud nejsou zadána žádná vyhledávací kritéria (například při prvním přechodu na tuto stránku), tak se zobrazí všechny zdroje, které v databázi jsou. Tabulka s výsledky má přitom omezený počet řádků a v případě překročení daného počtu se rozdělí na více stránek.

4.4.2 Přidávání zdroje

Stránka na přidávání zdrojů obsahuje formulář, který umožňuje vyplnit všechny informace o zdroji. Lze zde zvolit také typ zdroje. Podle tohoto typu se pak pomocí JavaScriptu a jQuery schovají pole, které pro zvolený typ nemají význam. Po stisku tlačítka na odeslání, se zdroj spolu s informacemi přidá do databáze, zobrazí se informace o úspěšné akci a po chvíli proběhne přesměrování, opět pomocí JavaScriptu, na stránku s detaily o nově přidaném zdroji.

Ke každému zdroji je také možnost přidat jeho rodiče. Pomocí vyhledávání popsaného v předchozí kapitole stačí požadovaného rodiče vyhledat a poté odkaz na něj vložit do připraveného pole při přidávání zdroje. Z tohoto odkazu je automaticky extrahován identifikátor daného zdroje, který se posléze uloží do databáze.

Vyplnit informace o zdroji lze ještě jedním způsobem – importováním citace v BibTeX formátu. K tomu je využita open source JavaScriptová knihovna *bibtexParseJs*. [1] Ta nabízí dvě metody – *toJSON()* pro převod BibTeX formátu na asociativní pole a *toBibtex()*, která dělá přesný opak. Stačí tedy zkopírovat požadovanrou citaci ve formátu BibTeX do připraveného pole ve formuláři a pomocí metody *toJSON()* se automaticky předvyplní nalezená pole.

4.4.3 Zobrazení informací o zdroji

Na této stránce se zobrazují všechny dostupné informace o zdroji. Zobrazují se zde v podstatě tři typy informací – bibliografické údaje, komentáře a soubory.

Část zobrazující bibliografické údaje obsahuje kromě tabulky se všemi uloženými informacemi také možnost exportování těchto dat ve formátu BibTeX. To je zajištěno pomocí stejné knihovny pro práci s tímto formátem, která byla popsána v předchozí kapitole. Po stisku tlačítka pro export se tedy pomocí metody *toBibtex()* vygeneruje a zobrazí požadovaná citace. Je zde také tlačítko pro editaci, pomocí které se lze dostat do obrazovku umožňující zobrazené informace měnit a poté uložit.

Pod bibliografickými údaji se vykresluje seznam komentářů spolu s textovým polem pro vkládání nových. Jak bylo uvedeno na začátku této sekce, vykreslování stránky je zajišťováno speciálním objektem, který je předáván do metody pro vykreslování v tomto modulu. Tento objekt umožňuje, kromě přímého přidávání HTML kódu do stránky, také vykreslovat wiki syntaxi. Toho je zde využito pro vykreslování komentářů, díky čemuž v nich lze používat různá formátování a při instalaci dalších rozšíření třeba i matematické výrazy. Zobrazené komentáře také obsahují tlačítka pro editaci a jejich mazání, pokud je aktuálně přihlášený uživatel jejich autorem.

Další zobrazovanou sekcí je seznam přiřazených souborů. Před generováním tohoto seznamu se vždy v databázi kontroluje, jestli je již možné soubor zobrazit. Zároveň je zde v případě dostatečných práv vygenerován formulář pro přidání nového souboru spolu s tlačítky pro mazání těch již existujících. Při nahrávání dokumentu se také kontroluje, zda jsou splněna všechna omezení daná správcem médií – tedy nepovolené znaky v názvu souboru nebo přímo nepovolené typy souborů. Zároveň je zjišťováno, jestli soubor s daným názvem v databázi již existuje.

Poslední částí, která se zobrazuje, je případný seznam odkazů na publikace, které mají uvedený aktuálně zobrazovaný zdroj jako jejich rodiče. Toto se hodí například, pokud máme v databázi obsažený jak celý sborník článku, tak jednotlivé články.

4.5 Zásuvný modul typu Action

Poslední částí tohoto rozšíření je třída *action_plugin_publications*, která je potomkem třídy *DokuWiki_Action_Plugin* popsané v kapitole 2.2.3. V metodě *register()* se zaregistruje odběr obou potřebných typů událostí a přiřadí se k nim patřičné obslužné metody.

4.5.1 Našeptávání

U každého editovatelného textového pole, které je generováno na nějaké stránce v tomto zásuvném modulu je implementováno našeptávání. Pro tyto účely bylo třeba využít technologie AJAX a funkce pro našeptávání v JavaScriptové knihovně jQuery UI.

V klientské části (tedy v JavaScriptu) se všem označeným prvkům, které lze editovat, aktivuje našeptávání pomocí metody *autocomplete()*. Pokud uživatel do nějakého takové pole začne psát, tak se pomocí metody *jQuery.post()* pošle na DokuWiki AJAX požadavek obsahující jméno databáze, jméno prvku a text, který uživatel již vyplnil.

Požadavek se poté na serveru zpracuje v této části rozšíření pomocí zachytávání událostí *AJAX_CALL_UNKNOWN*. V metodě pro obsluhu této události se nejdříve zkontroluje, že se skutečně jedná o požadavek o našeptávání. Poté se vyhledá patřičná databáze a v ní všechna požadovaná pole, která obsahují uživatelem již vyplněnou část. Seznam všech takových hodnot se poté odešle zpátky klientovi, který ho již jen automaticky vykreslí jako rozbalovací seznam.

4.5.2 Přístupová práva

U každého generovaného formuláře při zobrazování jakýchkoliv informací z databáze jsou kontrolována přístupová práva, a podle nich se zobrazuje dodatečná funkcionalita. K tomu je použita metoda *auth_quickaclcheck(id)*, která pro daný identifikátor v daném kontextu vrací nastavená práva. Ke každé databázi je přiřazen jmenný prostor a nastavení práv se čte právě z něj.

Jedinou výjimkou je zobrazování souborů, u kterých je třeba kontrolovat ještě jejich datum zveřejnění. Jelikož se soubory ukládají přímo do nastaveného jmenného prostoru, tak jsou normálně viditelné ve správci médií. Je tedy třeba zajistit, aby se i zde respektovalo natavení jejich data zveřejnění. Toho bylo docíleno pomocí zachytávání události *AUTH_ACL_CHECK*. V obslužné metodě se nejprve z konfiguračního souboru zjistí, jestli je ke kontrolovanému jmennému prostoru přiřazena v tomto doplňku nějaká databáze. Pokud ano, v databázi se pokusí vyhledat konkrétní soubor a v případě, že soubor v databázi existuje, navrací se práva podle jeho data zveřejnění. Tato kontrola přitom probíhá pouze pro uživatele s právem *read*. Pro uživatele s vyšším oprávněním jsou soubory vždy zobrazovány všechny.

Touto přímou změnou chování metody pro kontrolu práv je zajištěno jednotné zacházení s danými soubory napříč celou DokuWiki včetně případných doplňkových modulů.

Jelikož DokuWiki sama o sobě neumožňuje nastavování přístupových práv na jednotlivé soubory, tak se při jejich stahování nekontrolují práva přímo na dané soubory, ale pouze na jmenný prostor, ve kterém se nachází. A to i přesto, že pomocí zásuvných modulů tato práva na jednotlivé soubory nastavovat jdou. I když se totiž odkaz na soubor nikde nezobrazí, neoprávněný uživatel by jej stále mohl stahovat například pomocí ručního vypsání URL. Je tedy třeba zajistit, aby se daná práva kontrolovala alespoň pro soubory obsažené v naší databázi. Toho bylo docíleno pomocí zachytávání události *FETCH_MEDIA_STATUS*. V obslužné metodě se pomocí výše popsané metody pro kontrolu práv zjistí, jestli má současně přihlášený uživatel práva soubor stahovat. Pokud ne, je vrácen chybový kód.

4.6 Testování

Zásuvný modul byl vyvíjen pro DokuWiki verze 2014-09-29 s kódovým označením "Hrun". Testování probíhalo nejprve na domácím webovém serveru IIS (Internet Information Services) běžícím na platformě Windows s PHP verze 5.3.29. Na závěr byl doplněk testován na serveru eva.fit.vutbr.cz, který nabízí stejné prostředí jako wiki instalace výzkumné skupiny formálních modelů, pro kterou byl tento doplněk převážně vyvíjen.

Kapitola 5

Závěr

Cílem této práce bylo navrhnout a implementovat doplněk pro DokuWiki, který bude umožňovat spravování publikačních zdrojů. Vyvinutý zásuvný modul umožňuje přidávání či editování publikací v databázi, přičemž podporuje možnost spravovat takovýchto databází libovolné množství. Zároveň nabízí možnost přidané publikace komentovat a přiřazovat k nim soubory spolu s datem jejich zveřejnění.

To bylo také hlavní motivací pro vznik tohoto rozšíření. Výzkumná skupina formálních modelů pro předávání informací mezi sebou používá mimo jiné právě systém DokuWiki. Doplněk, který vzešel z této práce, jim umožní efektivněji sdílet vlastní práce či komentovat existující publikace.

Výsledný doplněk by se dal dále rozšiřovat. Jednou z možných funkcí, která se nabízí, je podpora pro více druhů citací mimo BibTeX – ať už pro exportování nebo importování. Například česká norma ISO ČSN 690 nebo různé další normy používané v zahraničí. Nabízí se třeba možnost propojení se servery jako `www.citace.com` nebo `citace.lib.vutbr.cz`. Žádný takovýto web ale v současnosti nenabízí nějaké aplikační rozhraní, kterého by se dalo využít. Stejně tak neexistuje žádná dostatečně funkční otevřená knihovna – ať už v jazyce PHP nebo v JavaScriptu.

Jiným možným rozšířením by byla možnost importovat publikace pomocí ISBN, ISSN či DOI. Toho by se dalo docílit využitím nějakého veřejného aplikačního rozhraní pro existující databáze publikací. Případně by také mohlo být dobré mít možnost importovat publikace ze seznamů na webových stránkách FIT VUT. [**?**]

Literatura

[1] bibtexParseJs. https://github.com/ORCID/bibtexParseJs.

[2] DokuWiki Plugin Development. https://www.dokuwiki.org/devel:plugins.

[3] PHP: Hypertext Preprocessor. http://www.php.net.

[4] Mahemoff, M.: *Ajax design patterns*. O'Reilly Media, Inc., 2006, iSBN 0-596-10180-5.

[5] PHPOffice: PHPExcel. https://github.com/PHPOffice/PHPExcel.

[6] WWW stránky: DokuWiki. https://www.dokuwiki.org.