



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Modul systému V3S pro práci s metadaty ze systému Crossref
Student:	Miroslav Kálal
Vedoucí:	Dr. Mgr. Jan Dvořák
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2019/20

Pokyny pro vypracování

Cílem práce je návrh a implementace modulu do systému V3S (interní systém ČVUT sloužící k evidenci publikační činnosti). Modul bude zajišťovat porovnání a aktualizaci dat o publikační činnosti mezi externím systémem Crossref a lokálním úložištěm systému V3S.

Dílní úkoly práce:

1. Seznamte se se systémem Crossref, zejména s API, které umožní přístup k datům. Seznamte se s relevantní částí úložiště dat, architekturou a implementační platformou systému V3S.
2. Společně s vedoucím práce přesněji specifikujte zadání. Držte se těchto hrubých požadavků:
 - modul bude pracovat v interaktivním módu, jehož rozhraní bude součástí systému V3S,
 - modul bude pracovat též v dávkovém módu.

Zadání dopřesněte i formou uživatelských scénářů práce s modulem.

3. Na základě specifikace proveďte návrh řešení a řešení implementujte. Držte se implementační platformy použité v systému V3S.

4. Implementovaný modul řádně otestujte, zdokumentujte a předejte zadavateli k nasazení do produkce.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 31. října 2018

Poděkování

Děkuji vedoucímu bakalářské práce za ochotu, trpělivost a cenné rady během celé tvorby práce.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů, zejména skutečnost, že České vysoké učení technické v Praze má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona.

V Praze dne 13. května 2019

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2019 Miroslav Kálal. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Kálal, Miroslav. *Modul systému V3S pro práci s metadaty ze systému Crossref*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2019.

Abstrakt

Předmětem práce je návrh a implementace nového modulu pro systém V3S. Tento systém slouží k evidenci publikační činnosti ČVUT v Praze. Modul rozšiřuje evidenci o stahování publikačních metadat z externího systému Crossref. Autorům nabízí automatické předvyplnění formuláře vytvoření záznamu daty ze systému Crossref. Tím urychluje a usnadňuje evidenční činnost autorů univerzity. Dále umožňuje porovnávání uložených záznamů s daty ze systému Crossref, díky tomu lze opravovat a aktualizovat záznamy. Modul byl implementován, úspěšně otestován a nasazen do provozu.

Klíčová slova Rozšíření webové aplikace, V3S, Crossref, evidence publikační činnosti ČVUT, CRIS, publikační metadata, REST, JSON, Jackson

Abstract

The work presents the design and the implementation of a new module for the V3S current research information system. The system serves for registration of publication activity of CTU in Prague. The module extends the registration with the download of publication metadata from the Crossref system. When the user enters a new DOI, it fetches the data from the Crossref system and

prefills the form to create a new record. Further, it enables comparison of the saved records with Crossref data, thanks to which the records can be periodically enhanced and updated. The module helps to speed up and simplify the registration activity of the university's authors. The module was implemented, successfully tested and deployed to production.

Keywords Web application extension, V3S, Crossref, CTU publishing activity registration, CRIS, publication metadata, REST, JSON, Jackson

Obsah

Úvod	1
1 Cíle práce	3
2 Analýza	5
2.1 Systém V3S	5
2.2 Digital Object Identifier	5
2.3 Aplikační rozhraní Crossref	6
2.4 Datový model	8
2.5 Uživatelské požadavky	9
2.6 Případy užití	10
3 Architektura a návrh řešení	13
3.1 Architektura a technologie V3S	13
3.2 Technologie Crossref API	15
3.3 Návrh řešení	15
4 Implementace	19
4.1 Mapování	19
4.2 Stahování metadat z Crossref API	20
4.3 Dávkový mód stahování metadat	21
4.4 Ukládání odpovědí Crossref API	22
4.5 Interaktivní mód	23
5 Testování	25
5.1 Jednotkové testy	25
5.2 Integrované testy	25
5.3 Testování před nasazením do provozu	26
6 Ostrý provoz	29

Závěr	31
Literatura	33
A Seznam použitých zkratk	37
B Obsah přiloženého CD	39
C Tabulky mapování Crossref JSON na V3S ExtPublication	41
D Dotaz na afiliace ČVUT v Crossref API	45

Seznam obrázků

2.1	Počet záznamů s DOI v průběhu času. Zdroj: dotaz na publikace do systému V3S.	7
2.2	Relativní počet záznamů s DOI oproti celkovému počtu. Zdroj: dotaz na publikace do systému V3S.	7
2.3	Datový model Crossref API (relevantní část pro systém V3S) . . .	8
2.4	Datový model externích publikací systému V3S (relevantní část pro systém Crossref)	9
3.1	Moduly systému V3S	14
4.1	Implementace dávkového módu	22

Seznam tabulek

C.1	Mapování	42
C.2	Mapování Autorů	42
C.3	Mapování typů	43

Úvod

Evidence vlastní publikační činnosti je v zájmu každé výzkumné organizace, nabízí přehled o její práci jako celku, jednotlivých součástí, pracovišť i jednotlivých výzkumníků. ČVUT jakožto výzkumně orientovaná univerzita má bohatou vědeckou a výzkumnou činnost, v jejímž rámci akademici i studenti publikují. Univerzita publikace svých autorů eviduje.

Výsledek této práce usnadní výzkumníkům zadávání informací o své publikační činnosti, zajistí jejich průběžnou aktualizaci a doplňování nezaevidovaných publikací. Motivací autora je praktické využití výsledků práce, které pomohou zjednodušit evidenční činnost výzkumníků.

Práce je rozdělena do 5 kapitol. Kapitola *Analýza* obsahuje popis systémů V3S a Crossref. Dále obsahuje popis rozhraní systému Crossref. Poslední částí *Analýzy* je seznam uživatelských požadavků a scénáře případů užití. Kapitola *Architektura a návrh* obsahuje popis technologií a architektury, na kterých je postaven systém V3S a rozhraní Crossref. Kapitulu uzavírá návrh řešení modulu. Kapitola *Implementace* popisuje samotné řešení modulu. Kapitola *Testování* obsahuje popis struktury testů a jejich výsledky. Poslední kapitola *Ostrý provoz* obsahuje poznatky z provozu modulu na produkci.

Cíle práce

Cílem práce je vyvinout modul rozšiřující systém V3S (interní systém ČVUT sloužící k evidenci publikační činnosti). Modul bude zajišťovat porovnávání a aktualizaci dat o publikační činnosti ČVUT s pomocí externího systému Crossref. Modul bude stahovat publikační metadata ze systému Crossref a ukládat je do lokálního úložiště systému V3S. Modul dále zajistí aktualizaci dříve uložených metadat. Uživatelům zajistí předvyplnění formuláře vkládání nového záznamu, pokud uživatel zná identifikátor DOI příslušné publikace.

Cílem analytické části práce je seznámení se systémem V3S, aplikačním rozhraním systému Crossref a identifikací digitálních objektů. Jeho součástí je popis architektury V3S a technologií potřebných k dosažení cíle praktické části. Posledním cílem je upřesnění zadání prostřednictvím seznamu uživatelských požadavků a případů užití modulu.

Cílem praktické části práce je návrh, implementace a otestování modulu. Dalším cílem je představení výsledků nasazení modulu do produkce.

Analýza

Tato kapitola obsahuje popis systému V3S, systému Crossref a jeho aplikačního rozhraní (API). Dále obsahuje popis datových modelů systému V3S a Crossref API. Kapitola uzavírá analýza uživatelských požadavků a případů užití nového modulu.

2.1 Systém V3S

V3S je interní systém ČVUT, který spadá do kategorie tzv. informačních systémů o aktuálním výzkumu (*Current Research Information Systems*) [1]. Byl vyvinut pro evidenci výsledků vědy (zejména publikačních) a aplikovaného výzkumu. „[...] V3S slouží k odesílání výsledků do RIV (Rejstřík informací o výsledcích), exportům pro statistické analýzy i k interním hodnocením vědeckovýzkumné činnosti.“ [2] Každý přihlášený uživatel má v systému V3S přístup k záznamům, kde je uveden mezi autory publikace a záznamům, které sám založil. Záznamy je možné vyhledávat a filtrovat, jednou použité filtry lze ukládat a opakovaně s nimi vyhledávat. Systém V3S v roce 2016 nahradil původní systém VVVS, který byl vyvíjen od 90. let 20. století. [3]

V3S je v rámci Informačního systému ČVUT propojen s mnoha dalšími komponentami. Zároveň spolupracuje s řadou externích systémů. [4] Z citačních databází WoS (*Web of Science*) a Scopus, do nichž si ČVUT předplácí přístup, získává záznamy o publikacích autorů ČVUT a údaje o citacích těchto publikací [5], [6].

2.2 Digital Object Identifier

DOI je standard pro identifikaci digitálních objektů. Může být přiřazen jakémukoli objektu, ať fyzickému, nebo virtuálnímu. Identifikátory jsou permanentní, po přiřazení k objektu je nelze změnit, nahradit ani přiřadit jinému objektu. Syntaxe každého identifikátoru se skládá ze dvou částí, oddělených

lomítkem. První část identifikuje vlastníka nebo organizaci, která DOI vydala, druhá část je samotný identifikátor, unikátní v rámci daného vlastníka/organizace. [7] Naopak umístění objektu se může změnit a tuto změnu lze reflektovat úpravou metadat v systému DOI, tím dojde k úpravě přesměrování.

DOI je ve sféře vědecké komunikace zavedeným standardem pro identifikaci vědeckých publikací. Jejich stále rostoucí význam pro publikace autorů z ČVUT dokumentují grafy na obrázcích 2.1 a 2.2.

2.3 Aplikační rozhraní Crossref

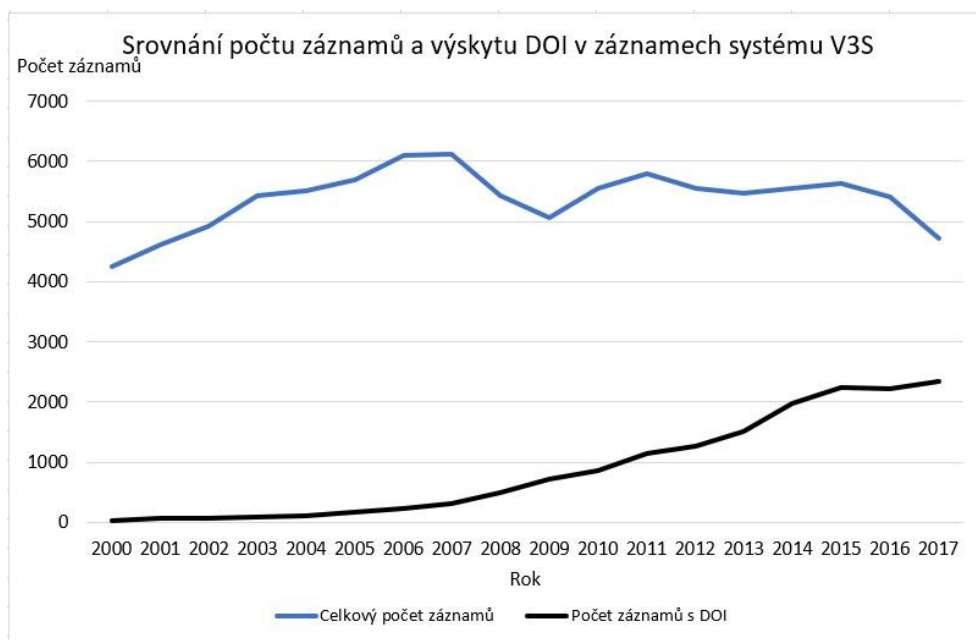
Crossref je nezisková organizace, která vytváří služby a nástroje, umožňující sdílení dat mezi odborníky [8]. Přístup k publikačním metadatům pro automatizované zpracování zajišťuje Crossref API přístupné z <https://api.crossref.org>. To nabízí více druhů metadat (členové Crossref, časopisy a další), modul pro V3S využívá pouze metadata publikací (*works*). API vrací jednotlivé záznamy, seznam záznamů nebo jen hlavičky jako informaci, zda se hledaný záznam vyskytuje v systému Crossref.

Přes Crossref API lze k jednotlivým publikačním metadatům přistupovat pomocí identifikátoru DOI. Pokud není známý, lze pokládat dotazy (*query*) podle různých kritérií, například:

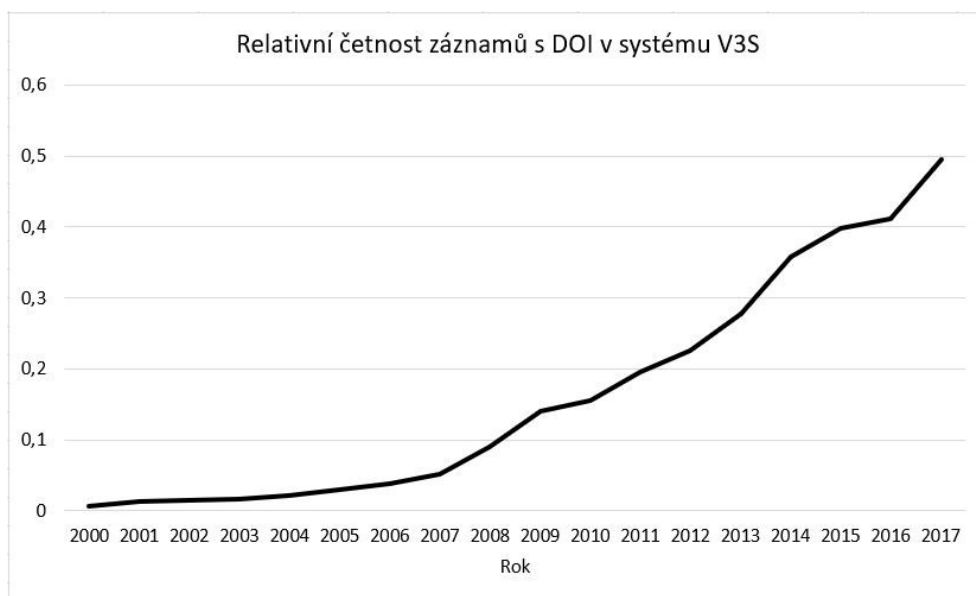
- title (název),
- container-title (název publikace, ve které se nachází výsledek, např. název časopisu, sborník, knižní řada),
- author (jméno autora),
- editor (jméno editora),
- translator (jméno překladatele),
- contributor (příspěvatel – společný pojem pro autora, editora a překladatele),
- bibliographic (bibliografické informace o publikaci – názvy publikace, autoři, ISSN a nebo rok publikování),
- affiliation (afiliace¹ příspěvatele).

Každý dotaz vrací seznam publikačních metadat. Výsledky dotazů lze dále zúžit pomocí filtrů. Pro publikační metadata nabízí Crossref API celkem 66 filtrů. V jednom požadavku se může nacházet více filtrů, API vrátí jen ty

¹neformálně definovaný pojem, obecně znamená propojení s organizací, politickou stranou nebo náboženstvím. [9]. V kontextu vědecké komunikace jde o instituci, na které autor prováděl výzkum, jehož výsledky popisuje v publikaci.

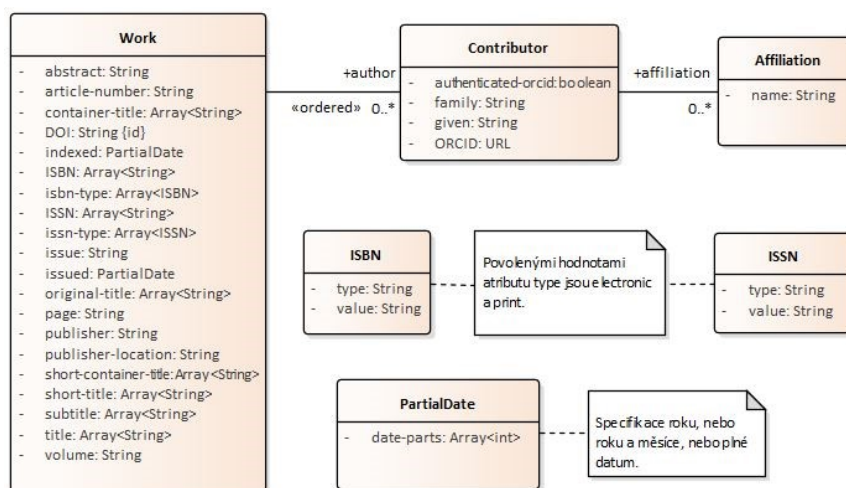


Obrázek 2.1: Počet záznamů s DOI v průběhu času. Zdroj: dotaz na publikace do systému V3S.



Obrázek 2.2: Relativní počet záznamů s DOI oproti celkovému počtu. Zdroj: dotaz na publikace do systému V3S.

2. ANALÝZA



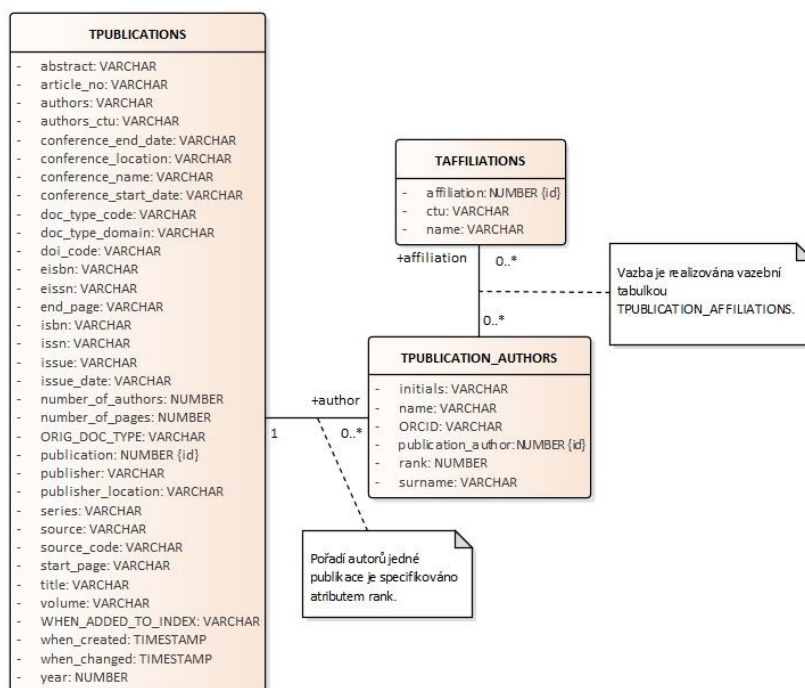
Obrázek 2.3: Datový model Crossref API (relevantní část pro systém V3S)

výsledky, které vyhovují všem zadaným filtrům. Pokud se zadá jeden filtr vícekrát, API vrátí výsledky, z nichž každý vyhovuje alespoň nebo právě jednomu z těchto filtrů. Pro optimalizaci vrácených metadat se do požadavku přidá parametr obsahující názvy všech atributů, které se mají v metadatech objevit. [10]

2.4 Datový model

Diagramy na obrázcích 2.3 a 2.4 ukazují struktury datových modelů systému V3S, resp. Crossref API. Je z nich patrné, že modely si do značné míry odpovídají.

Rozdíly jsou například v uchování identifikátorů ISBN a ISSN. Ty je nutné do systému V3S přemapovat podle typu do zvláštních údajů isbn (tištěný), eisbn (elektronický), resp. issn, eissn.



Obrázek 2.4: Datový model externích publikací systému V3S (relevantní část pro systém Crossref)

2.5 Uživatelské požadavky

K upřesnění zadání slouží uživatelské požadavky, dělí se na funkční a nefunkční. Implementační část se zabývá plněním následujících funkčních požadavků:

F1 stahovat metadata publikací ze systému Crossref na základě:

- F1.1 DOI publikace,
- F1.2 afiliace autora k ČVUT,

F2 předvyplnit formulář vkládání nového záznamu při vyplnění DOI uživatelem,

F3 periodicky stahovat metadata v dávkovém módu na základě:

- F3.1 DOI publikace,
- F3.2 afiliace autora k ČVUT.

Dále se implementace zabývá plněním nefunkčního požadavku:

- N1 Získané odpovědi API ukládat do systému pro správu verzí pro ladění, řešení případných nejasností a ověření, kdy byla metadata v konkrétním tvaru stažena.

2.6 Případy užití

Detailní specifikaci uživatelských požadavků tvoří případy užití, popsané v následujících uživatelských scénářích.

2.6.1 UC1 Stažení a zpracování metadat ze systému Crossref

1. Systém V3S pošle požadavek systému Crossref s vybraným DOI.
2. Systém Crossref vrátí metadata publikace s odpovídajícím DOI.
3. Systém V3S metadata zformátuje a uloží v systému souborů do verzovacího systému.
4. Systém V3S metadata zpracuje a uloží do databáze jako externí publikaci.

2.6.2 UC2 Vložení záznamu o publikaci do V3S

Předpoklad: Uživatel je přihlášený v systému V3S.

1. Uživatel vybere v přehledu možnost vložit na základě DOI.
2. Systém V3S umožní uživateli zadat DOI.
3. Uživatel vyplní DOI zadávané publikace.
4. *Include* UC1 (Stažení a zpracování metadat ze systému Crossref), systém V3S zahájí zakládání záznamu, předvyplní formulář založení záznamu, vybere typ publikace na základě typu v metadatech.
5. Systém V3S přeměruje Uživatele na formulář vkládání nového záznamu, ten předvyplní (typ publikace odvodí od typu záznamu externí publikace) a umožní Uživateli editaci formuláře.
6. Uživatel zkontroluje a doplní údaje ve formuláři a potvrdí založení záznamu.
7. Systém V3S vyplněné údaje uloží jako nový výsledek.

2.6.3 UC3 Dávkové stažení metadat podle DOI

1. V nastavený čas se spustí dávkový mód systému V3S pro stahování metadat ze systému Crossref na základě DOI publikací.
2. Systém V3S začne stahovat metadata na základě DOI publikací v databázi V3S, upřednostní ta, která ještě nebyla sklizena a poté ta, která byla sklizena před nejdelší dobou.
3. Pro každé DOI: *Include UC1* (Stažení a zpracování metadat ze systému Crossref).

2.6.4 UC4 Dávkové stažení metadat podle afiliace k ČVUT

1. V nastavený čase se spustí dávkový mód Systému V3S pro stahování metadat ze systému Crossref na základě afiliace autorů k ČVUT.
2. Systém V3S pošle požadavek s afiliací systému Crossref.
3. Systém Crossref vrátí zprávu obsahující metadata všech publikací s autory, jejichž afiliace odpovídá zadané.
4. Systém V3S data zpracuje a pro každou publikaci založí nový záznam, nebo aktualizuje již existující.

Architektura a návrh řešení

Tato kapitola obsahuje popis architektury systému V3S, rozhraní Crossref API a technologií nezbytných k dokončení praktické části. Dále obsahuje návrh řešení požadavků uvedených v sekci 2.5.

3.1 Architektura a technologie V3S

Kód aplikace V3S je veden v systému pro správu verzí Git [23] s repozitářem na VIC ČVUT v Praze. Kód je veden v projektu *vvvs*, jehož strukturu zachycuje diagram na obrázku 3.1. Projekt se skládá ze 7 modulů. Modul V3S obsahuje konfiguraci a slouží ke správě zdrojových souborů aplikace V3S ve verzovacím systému. Všechny zbývající moduly jsou jeho součástí.

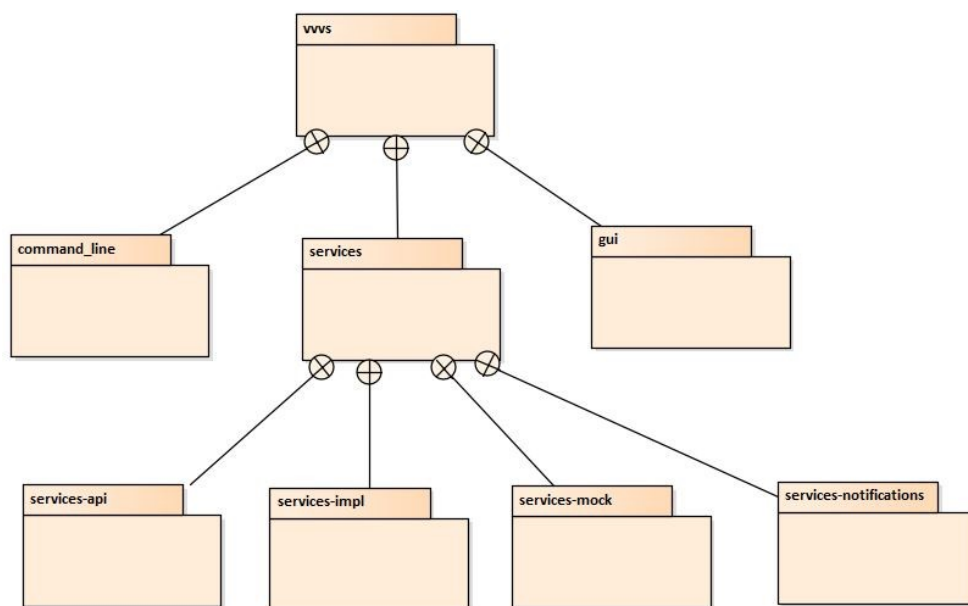
Jádrum implementace V3S je modul *Services*, rozdělený na 4 dílčí moduly, popisující rozhraní (*Services-api*), implementaci (*Services-impl*, *Services-mock*) a modul *Services-notifications*. Pro testování je určen modul *Services-mock*, který nahrazuje skutečnou implementaci *Services-impl*. V Modulu *Services* je logika rozdělena do nezávislých služeb. Modul *Services-notifications* umožňuje odesílání emailů uživatelům.

Modul *command_line* obsahuje rozhraní příkazové řádky pro nezávislé spuštění dávkových úloh.

Část aplikační logiky aplikace V3S je realizována pomocí databázových pohledů, uložených skriptů a procedur v databázi. Jejich zdrojové kódy jsou vedeny ve vlastním projektu *vvvs-db*. Úpravy databáze jsou popsány pomocí změnových skriptů a do databází jsou promítány pomocí nástroje *Liquibase*. [11]

Aplikace V3S je implementovaná v hybridním, objektově orientovaném programovacím jazyce Java ve verzi 1.7 [12]. Systém běží v aplikačním kontejneru *frameworku* Spring, který nabízí nástroje jako je převzetí kontroly nad vytvářením objektů. Konfigurace *frameworku* Spring a dalších součástí se provádí pomocí souborů ve formátu XML. [13]

3. ARCHITEKTURA A NÁVRH ŘEŠENÍ



Obrázek 3.1: Moduly systému V3S

XML (*eXtensible Markup Language*) je značkovací jazyk, který popisuje třídu datových objektů XML dokumenty (*XML documents*). Každý dokument reprezentuje entitu obsahující data. Tato data se dělí na znaky obsahující informace a na vyznačení logické struktury dat. [14]

Modul Gui aplikace V3S implementuje webové uživatelské rozhraní pomocí JSF (*Java Server Faces*). JSF je technologie určená pro vytváření webových rozhraní. Umožňuje přístup k datům a logice na straně *serveru*. Rozhraní implementované pomocí JSF se skládá z komponent, které jsou podporovány objekty jazyka Java. Takové objekty jsou nazývány *Backing beans*. [15]

Aplikace V3S využívá technologii Jackson k převádění dat mezi formátem JSON a objekty jazyka Java. K tomu slouží třída *ObjectMapper* z modulu *data-bind*. [19] JSON je textový formát odvozený od zápisu objektů v programovacím jazyce Javascript. JSON slouží k textové reprezentaci strukturovaných dat. [22]

Úložištěm aplikace V3S je relační databáze Oracle. Databáze je uspořádaná soustava dat. Relační databáze strukturuje uložená data do tabulek. Jeden záznam odpovídá jednomu řádku tabulky, jeden sloupec může být společný pro více tabulek, čímž vyjadřují vztah (relaci) mezi tabulkami. Pro práci s relační databází slouží jazyk SQL (*Structured Query Language*). [16]

K ukládání objektů do databáze používá aplikace V3S framework MyBatis. Ten umožňuje mapovat data na tabulky relačních databází. K tomu používá

konfigurační soubory (*Mapper*), ve kterých se nastaví příslušné SQL dotazy a definují mapování datových objektů na tabulky relačních databází a naopak. [17] Aplikace V3S načítá údaje z databáze zásadně přes vrstvu pohledů, které databáze V3S pro aplikaci obsahuje. Úpravy dat v databázi pak aplikace provádí vždy voláním příslušných uložených procedur a funkcí členěných do balíků podle jednotlivých entit. Databáze mj. realizuje logiku vyhodnocování uživatelských oprávnění.

Databáze je společná s aplikací EZOP, která eviduje výzkumné i další projekty řešené na ČVUT. EZOP obsluhuje proces návrhu a interního schvalování projektu na ČVUT a v případě získání financování pak proces administrace tohoto projektu. V aplikaci V3S se výsledky navazují na ty projekty, při jejichž řešení bylo těchto projektů dosaženo.

Překlad a sestavování aplikace V3S se provádí pomocí nástroje Maven [18]. Sestavení pro nasazení na servery se provádí v nástroji pro průběžnou integraci Jenkins.

Průběžná integrace (*Continuous integration*) je způsob, jak rychle poskytovat zpětnou vazbu při vývoji softwaru, monitorovat, testovat a analyzovat jeho jednotlivé verze. Jenkins umožňuje automatizované sestavování aplikací, pravidelné i interaktivní spouštění různých úloh aj. [20]

3.2 Technologie Crossref API

Crossref API je postaveno jako REST API. REST (*REpresentational State Transfer*) je návrh architektury, který v případě webových služeb rozděluje služby na jednotlivé datové zdroje, ke kterým lze přistupovat. [21]

API zpřístupňuje jednotlivé typy dat pomocí REST zdrojů (*resources*). Nový modul využívá pouze zdroj works. Jednotlivé publikace identifikuje pomocí jejich DOI. Údaje o publikaci s konkrétním DOI lze získat zasláním požadavku HTTP GET na adresu `https://api.crossref.org/works/{DOI}`, kde se za DOI dosadí vlastní DOI, přičemž lomítka v něm není třeba nahrazovat zástupnou sekvencí `%2f` (odchylka od pravidel REST).

Crossref API též umožňuje dotazování dle mnoha kritérií, viz sekce 2.3. Dotaz na výběr publikací podle afiliace jejich autorů k ČVUT (požadavek F3.2) je uveden v příloze D. Vyplněnost údaje o afiliaci je v databázi Crossref nedostatečná, blíže viz sekce 5.3.

API vrací odpovědi v textovém formátu JSON.

3.3 Návrh řešení

Návrh tříd dodržuje architekturu systému V3S. Rozšiřuje již existující uživatelské rozhraní (GUI) a způsob spouštění dávkového módu z příkazové řádky (provádí systém Jenkins).

3.3.1 Mapování

Přehled mapování jednotlivých atributů ukazuje tabulka C.1 v příloze C. Z mapování byl vynechán atribut `series`, který popisuje knižní řady apod. V metadatech z Crossref API se tato informace může vyskytnout v atributu `container-title`, není ovšem jasné, jak rozlišit případy, kdy se jedná o informaci mapovanou do atributu `series` a kdy do atributu `source`.

Zároveň je třeba mapovat samotný údaj typ externí publikace. Toto mapování je prezentováno v tabulce C.3. Rovněž je nutné mapovat jednotlivé autory publikací (tabulka C.2).

3.3.2 Dávkový mód

V současné implementaci V3S se dávkový mód spouští v pravidelných nočních sestavách zajištěných systémem Jenkins. V nastavený čas provede sestavení aplikace, její přenos na server a její spuštění na serveru. Požadované synchronizace se volí parametry příkazové řádky. Nově byly přidány volby `-cd` (`—synchronizeCrossrefAllExistingRecordsByDoi`) a `-ca` (`—synchronizeCrossrefAllExistingRecordsByAffiliation`).

Pro spuštění dávkového módu stahování z Crossref API slouží třída `CrossrefCtuRecordsSynchronizationJob`. Ta obsahuje metody `synchronizeAllRecordsByDoi` a `synchronizeAllRecordsByAffiliation`, které spouštějí samotnou logiku módu vyhledávání podle DOI, resp. afiliace. Třída má jeden vnitřní atribut s číselným identifikátorem uživatele, který stahování spustil. V případě dávkového spuštění se používá hodnota 0, která reprezentuje samotnou aplikaci V3S.

Strukturu řešení ukazuje diagram na obrázku 4.1. Přes rozhraní `CrossrefSynchronizationService` se spouští dávkové zpracování podle toho, zda se jedná o stahování na základě kódu DOI nebo afiliace autorů publikace.

Stahování metadat z Crossref API zajišťuje rozhraní `CrossrefClient`. Rozhraní umožňuje posílat žádosti o publikační metadata vybraná pomocí DOI (metoda `retrieveByDoi`). Dále umožňuje výběr pomocí autorské afiliace (metoda `retrieveByAffiliation`), alternativně s možností specifikace atributů, které Crossref vrátí (metoda `retrieveFieldsByAffiliation`).

3.3.3 Ukládání a verzování stažených metadat

Komunikaci lze realizovat prostřednictvím rozhraní třídy `RestTemplate` z frameworku Spring. Požadavek N1 vyžaduje ukládání odpovědí Crossref API. Řešení implementuje třída `CrossrefJsonFileRecordingRestTemplate`, rozšiřující třídu `RestTemplate`. Třída, stejně jako `RestTemplate`, umožňuje komunikaci s REST službami a konverzi příchozích odpovědí. Před konverzí do výstupního typu a předáním odpovědi provede uložení odpovědi do souboru v textové podobě. Pro lepší čitelnost je obsah odpovědí rozšířen o řádkování

a odsazování jednotlivých atributů. Třída ukládá soubory do verzovacího systému Git. Příkaz pro vložení do verzovacího systému (*commit*) je volán pro každý soubor zvlášť.

Třída `CrossrefJsonFileRecordingRestTemplate` obsahuje metodu `init`, která nastavuje vnitřní datové struktury instance třídy a v případě potřeby i repozitář Git a instanci třídy `CrossrefJsonFileRecordingRootElementHttpMessageConverter`, která zajišťuje samotné ukládání. Nastavení ukládání proběhne pouze pokud je v konfiguraci nastavena vlastnost (*property*) `import.sync.crossref.saveResponseToFiles`. Další vlastnost, `import.sync.crossref.outputPath`, určuje umístění repozitáře. Metoda `init` se volá vždy po vytvoření instance třídy `CrossrefJsonFileRecordingRestTemplate`.

3.3.4 Interaktivní mód

Interaktivní mód je přístupný z obrazovky přehledu (úvodní obrazovka aplikace V3S, přístupná přihlášeným uživatelům) přes odkaz „vložit nový záznam podle DOI“. K zadání kódu DOI slouží obrazovka `createByDoi`. Ta obsahuje jediné pole pro zadání kódu a tlačítko „Vložení dle kódu DOI“.

S obrazovkou je spojena třída `CreateResultFromDoiBean`, rozšiřující třídu `BackingBean`. Třída obsahuje metodu `insertByDoiCode`, ve které se z Crossref API metadata stáhnou, namapují a uloží do databáze.

Po uložení do databáze probíhá přesměrování na obrazovku `create`. S ní je spojena třída `ResultBean` aplikace V3S. Ta je rozšířena o metodu `newRecordFromDoiCode`, která zajišťuje konverzi uložených metadat do instance třídy `ResultWithAuthorsAndAffiliations` aplikace V3S, se kterou pracuje již implementovaný formulář vkládání nového záznamu. Metoda má argumenty `doiCode` a `hasRight` (určuje, zda má uživatel právo na provádění operace, implementované v této metodě).

Implementace

Tato kapitola obsahuje popis řešení modulu. Zvláště popisuje část mapování publikačních metadat a jednotlivých požadavků.

4.1 Mapování

Pro mapování metadat slouží třída `CrossrefJsonToPublicationMapper` (dále jen *mapper*). Má jedinou veřejně přístupnou metodu `map`, která přijímá odpověď Crossref API ve formátu JSON ve formě textového řetězce (instance třídy `String`) a vrací instanci třídy `ExtPublication` (slouží k vnitřní reprezentaci publikačních metadat) s namapovanými daty. Mapper využívá třídu `ObjectMapper` z modulu `data-bind` (Jackson), kterým převede vstupní řetězec na instanci třídy `JsonNode` (Jackson). Tato třída umožňuje práci s jednotlivými atributy, jejich vyhledávání podle názvu atributu nebo jeho pořadí mezi atributy instance. Atributy lze extrahovat jako další instance `JsonNode`, seznam instancí, jako některý z primitivních typů jazyka Java nebo instance třídy `String`.

S instancí `JsonNode` pracuje mapper přes rozhraní třídy `JsonNodeTravelingExtractor`, která obaluje instanci `JsonNode` a rozšiřuje možnosti transformace atributů. V případě, že se v datech nevyskytuje hledaný atribut, vrací třída `JsonNodeTravelingExtractor` instanci třídy `NullNode` (Jackson), která reprezentuje hodnotu typu *null* formátu JSON. Výjimkou je transformace na textový řetězec, kdy třída `JsonNodeTravelingExtractor` vrací prázdný řetězec.

První se mapují povinné atributy, pokud některý z nich nelze namapovat, nebo při mapování dojde k chybě, vrátí metoda `map` hodnotu `null`.

Třída `CrossrefPublicationTypeMapping` implementuje mapování typů metadat Crossref API na typy externích publikací aplikace V3S. Obsahuje instanci mapy, ve které jsou uloženy hodnoty typů. Klíče v mapě jsou typy Crossref API, hodnoty (*values*) jsou typy aplikace V3S implementované jako

výčty `SourceType` a `DocType`. Mapování typů popisuje tabulka C.3 v příloze C.

4.2 Stahování metadat z Crossref API

Třída `CrossrefClientImpl` implementuje rozhraní `CrossrefClient`. Umožňuje komunikaci s Crossref API pomocí 3 metod (požadavky F1.1, F1.2).

Metoda `retrieveByDoi` má jediný argument *doi*, jeho hodnota se vloží do URL dotazu, který se předá privátní metodě `retrieve`.

Metoda `retrieve` předá hodnotu argumentu *url* instanci třídy `CrossrefJsonFileRecordingRestTemplate` přes metodu `getForEntity` (implementovaná třídou `RestTemplate` [13]). Druhým argumentem je typ, který se vrátí v obalující, generické třídě `ResponseEntity`. Jako typ se používá třída `String`.

Třída `AddUserContactHttpRequestInterceptor` přidává hlavičku `User-Agent` do požadavků, díky čemuž jsou rychleji obslouženy – Crossref nabízí rychlejší zpracování pro uživatele, kteří se identifikují. [10] Instance třídy je vytvořena a vložena do seznamu instance třídy `CrossrefJsonFileRecordingRestTemplate` pomocí konfigurace frameworku Spring.

Metoda `retrieveByAffiliation` předá hodnotu svého argumentu *affiliation* metodě `retrieveFieldsByAffiliation`, která umožňuje výběr atributů publikačních metadat, které vrátí Crossref API. K tomu slouží druhý argument *fields* (prvním argumentem je shodně s metodou `retrieveByAffiliation` *affiliation*). Do *fields* vkládá metoda `retrieveByAffiliation` hodnotu konstanty `SELECTED_ALL_FIELDS`, která obsahuje prázdný textový řetězec. Pokud je ve *fields* prázdný textový řetězec, metoda do URL dotazu nevkládá příkaz `select`. Jinak je tento příkaz vložen spolu s hodnotou *fields*. Metoda předá sestavený URL dotaz metodě `retrieve`.

Třída `ExternalPublicationServiceImpl` aplikace V3S slouží ke komunikaci s externími systémy. Je rozšířena o 3 následující metody. Metoda `findRemoteByCrossrefDoi` má jediný argument *doi*. Metoda stahuje publikační metadata přes rozhraní `CrossrefClient` (předá argument *doi*) a poté je mapuje pomocí instance třídy `CrossrefJsonToPublicationMapper`. Metoda vrací instanci třídy `ExtPublication` nebo hodnotu `null`, pokud selhalo mapování.

Metoda `insertOrUpdateByDoi` má dva argumenty – *publication* (instance třídy `ExtPublication`) a *userId* (identifikátor uživatele). Metoda aktualizuje záznam v databázi se stejným DOI jako *publication*, nebo vytvoří nový záznam.

Metoda `findRemoteByCrossrefAffiliation` stahuje publikační metadata pomocí metody `retrieveFieldsByAffiliation` třídy `CrossrefClientImpl`. Metodě předá jako první argument textový řetězec obsahující několik variant afiliace k ČVUT. Jako druhý argument předá text „DOI“. Metoda `retrieveFieldsByAffiliation` pak vrátí zprávu s hodnotami DOI záznamů publikací, které obsahují autora s alespoň jednou ze zadaných afiliací. Poté provede konverzi pomocí třídy

ObjectMapper na instanci třídy `JsonNode`, ze které extrahuje jednotlivá DOI. Pro každé DOI volá metodu `retrieveByDoi` třídy `CrossrefClientImpl` a vrácené odpovědi nejdříve mapuje pomocí instance třídy `CrossrefJsonToPublicationMapper`. Úspěšně namapované odpovědi (instance třídy `ExtPublication`) metoda ukládá do seznamu. Metoda `findRemoteByCrossrefAffiliation` vrací seznam instancí třídy `ExtPublication`.

4.3 Dávkový mód stahování metadat

Dávkový mód stahování metadat popisuje diagram na obrázku 4.1. Je spouštěn metodami třídy `CrossrefCtuRecordsSynchronizationJob`. Ta má dvě veřejně přístupné metody. Metoda `synchronizeAllRecordsByDoi` volá metodu `synchronizeAllCtuRecordsByDoi` třídy `CrossrefSynchronizationServiceImpl` (požadavek F3.1). Metoda `synchronizeAllRecordsByAffiliation` volá metodu `synchronizeAllCtuRecordsByAffiliation` téže třídy (požadavek F3.2).

Třída `CrossrefSynchronizationServiceImpl` implementuje rozhraní `CrossrefSynchronizationService`. Třída dávkově zpracovává všechna DOI, která vrací třída `ExternalServiceImpl` metodou `synchronizeAllCtuRecordsByDoi`.

Metoda `synchronizeAllCtuRecordsByDoi` volá metodu `findForCrossrefToSynchronize`, třídy `ExternalPublicationServiceImpl`, která vrací seznam instancí třídy `Doi`. Třída `Doi` obsahuje pouze DOI kód záznamu v databázi, používá se jako typ položky v seznamu, který vrací SQL dotaz uložený v konfiguračním souboru. Dotaz provádí `MyBatis`.

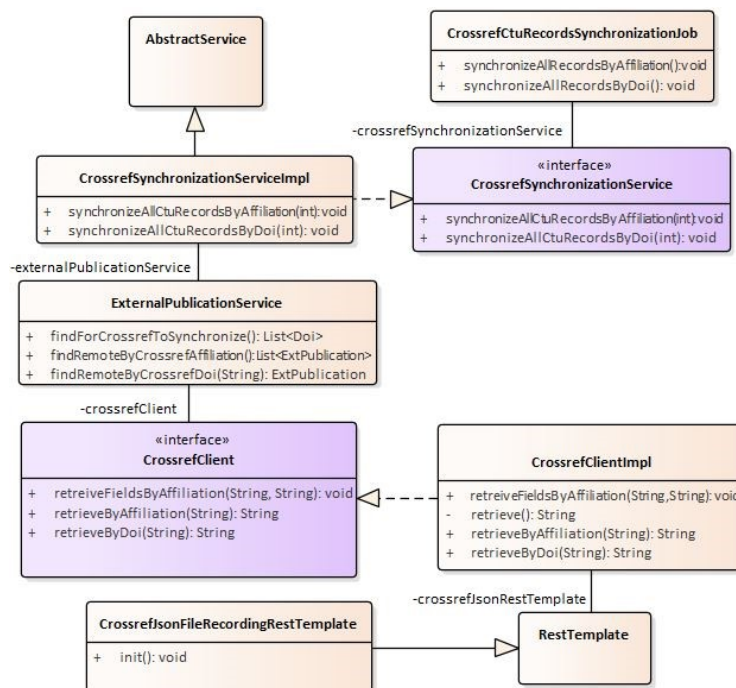
Metoda `synchronizeAllCtuRecordsByDoi` zpracovává postupně celý seznam. Pro každou položku (instanci třídy `Doi`) volá metodu `publicationByDoi`, které položku předá. Následně vrácenou instanci třídy `ExtPublication`, nebo hodnotu `null`, předá metodě `savePublication`. Tím končí zpracování jedné položky. Metoda končí po zpracování všech položek.

Metoda `publicationByDoi` má jediný argument `doi` (instance třídy `Doi`). Metoda volá `findRemoteByCrossrefDoi` třídy `ExternalPublicationServiceImpl`, které předá kód DOI z argumentu pomocí metody `getDoi`. Metoda vrací instanci třídy `ExtPublication`.

Metoda `savePublication` má dva argumenty, `publication` a `userId`. Metoda ukládá publikační metadata v `publication` pomocí metody `insertOrUpdateByDoi` třídy `ExternalPublicationServiceImpl`.

Metoda `synchronizeAllCtuRecordsByAffiliation` volá metodu `findRemoteByCrossrefAffiliation` třídy `ExternalPublicationServiceImpl` a zpracovává postupně seznam vrácených publikací (instance `ExtPublication`). Každou publikaci uloží do databáze pomocí metody `savePublication`.

4. IMPLEMENTACE



Obrázek 4.1: Implementace dávkového módu

4.4 Ukládání odpovědí Crossref API

Třída `CrossrefJsonFileRecordingRestTemplate` rozšiřuje třídu `RestTemplate` (Spring) o inicializační metodu `init`. Metoda se volá vždy po vytvoření instance třídy `CrossrefJsonFileRecordingRestTemplate` (pomocí anotace `PostConstruct`). Metoda přidá instanci třídy `CrossrefJsonFileRecordingRootElementHttpMessageConverter` (dále jen *converter*) do seznamu `messageConverters` třídy `RestTemplate` přes metodu `getConverters`. Converter zajišťuje konverzi odpovědí Crossref API v případě, že jako výstupní typ je zvolena třída `String`. V opačném případě přenechává konverzi původní implementaci třídy `RestTemplate`. Dále zajišťuje ukládání zpráv do repozitáře Git (požadavek N1).

Před uložením do verzovacího systému Git je každá zpráva doplněna o řádkování a odsazení jednotlivých JSON atributů pro lepší čitelnost. Poté se každá zpráva uloží do vlastního souboru. Názvem takto uloženého souboru je DOI zpráva s nahrazenými nealfanumerickými znaky. Toto nahrazování implementuje třída `StringEncoder`, která využívá třídu `UriUtils` z frameworku Spring [13]. Poté probíhá uložení do verzovacího systému (`commit`), pokud se do repozitáře přidávají změny (nový záznam, modifikace již existujícího).

4.5 Interaktivní mód

Obrazovka `createByDoi` umožňuje zadat kód DOI a po stisku tlačítka „Vložení dle kódu DOI“ se zavolá metoda `setDoiCode`, která uloží hodnotu kódu DOI do privátního atributu instance třídy `CreateResultFromDoiBean` (třída rozšiřuje abstraktní třídu `BackingBean` aplikace V3S). Poté se spustí metoda `insertByDoiCode`, nejdříve ověří, zda byl zadán validní kód DOI – neprázdný, odpovídající syntaxi DOI (viz sekce 2.2). Ke kontrole struktury slouží regulární výraz.

Pokud je DOI v pořádku, pokračuje metoda `insertByDoiCode` stažením a mapováním metadat z `Crossref API`. To zajišťuje rozhraní `ExternalPublicationService`, kterou metoda vybere z instance třídy `Services` (vnitřní atribut třídy `BackingBean`) voláním metody `getExternalPublicationService`. Třída `Services` obsahuje všechny služby, které se v GUI využívají. Toto rozhraní implementuje třída `ExternalServicesImpl` (dále jen služba). Metoda `insertByDoiCode` volá metodu `findRemoteByCrossrefDoi` (metoda služby) a předává jí atribut `doiCode` třídy `CreateResultFromDoiBean` jako argument. Metoda vrací instanci třídy `ExtPublication` systému V3S s metadaty z `Crossref API`. Poté metoda `insertByDoiCode` zavolá metodu `insertOrUpdateByDoi` (metoda služby), které předá instanci třídy `ExtPublication` a číslo přihlášeného uživatele (uložené ve zděděném vnitřním atributu třídy `BackingBean`). Metoda `insertOrUpdateByDoi` uloží metadata do databáze systému V3S.

Metoda `insertByDoiCode` vrací textový řetězec, který způsobí přesměrování na obrazovku `create` (zajišťuje JSF), která obsahuje formulář pro vložení nového záznamu. Metoda `newRecordFromDoiCode` rozšiřuje třídu `ResultBean` aplikace V3S, kterou obrazovka `create` využívá. Metoda vybere z databáze V3S záznam s DOI, které je v argumentu `doiCode`. Pokud záznam existuje, dojde k vyplnění formuláře daty ze záznamu (požadavek F2). V opačném případě dojde k přesměrování uživatele na obrazovku `createByDoi`.

Testování

Tato kapitola obsahuje popis testů a jejich výsledky. Testy jsou rozdělené na jednotkové, integrační a testy na testovacím prostředí. Pro jednotkové (*unit*) a integrační testy jsou automatizované, pro jejich realizaci se používá JUnit verze 4 [24]. Jednotkové testy byly po zvážení účelnosti oproti nákladům zpracovány pouze pro třídu `CrossrefJsonToPublicationMapper`.

5.1 Jednotkové testy

V testech třídy `CrossrefJsonToPublicationMapper` se vždy testuje výsledek, který vrátí metoda `map`.

Testuje se mapování namátkou vybraných metadat z Crossref API, které má skončit vrácením instance třídy `ExtPublication` s korektně namapovanými atributy. Dále se testuje mapování zprávy, kterou vrací Crossref API při nenalezení záznamu. V takovém případě má třída vrátit hodnotu `null`. Dále se testuje mapování metadat, ve kterých chybí povinná pole a testování rozlišování bibliografických identifikátorů ISBN a ISSN. Testuje se rozlišování, zda jde o elektronický nebo tištěný identifikátor. Na závěr se testuje mapování umělého záznamu obsahujícího všechny v současnosti mapovatelné atributy.

Všechny testy jsou implementovány ve třídě `CrossrefJsonToPublicationMapperTest`. Po odladění prochází třída `CrossrefJsonToPublicationMapper` všemi testy. Tyto test se spouští v rámci a také vždy při sestavování aplikace pro nasazení.

5.2 Integrační testy

Integrační testy slouží k testování implementace celých funkčních celků a komunikace s Crossref API. Všechny integrační testy tedy vyžadují spojení s Crossref API. Dále jsou všechny testy implementované v oddělených třídách. Název každé testovací třídy končí zkratkou `IT` – integrační test. Každá tes-

testovací třída rozšiřuje třídu `AbstractIT`, která nastaví konfiguraci frameworku Spring – nastavení testovacího profilu a spouštění JUnit.

Třída `CrossrefClientImpl` umožňuje stahování metadat z Crossref API. Její testovací třídou je `CrossrefClientIT`. Metoda `retrieveByDoi` se testuje na vrácení neprázdné hodnoty pro namátkou vybraný kód DOI. Tato metoda se dále testuje na vrácení neprázdné hodnoty s kódem DOI vedoucím k velkým metadatům. Cílem je ověřit, zda třída korektně vrací i velké záznamy. Metoda `retrieveByAffiliation` se testuje na vrácení neprázdné hodnoty, která obsahuje atribut afiliace s hodnotou CTU. Třída `CrossrefClientImpl` prochází všemi testy.

Třída `CrossrefJsonFileRecordingRestTemplate` rozšiřuje třídu `RestTemplate` o nastavení ukládání odpovědí Crossref API do repozitáře. Dále do každého dotazu přidává hlavičku s kontaktními informacemi systému V3S. Její testovací třídou je `CrossrefRestTemplateIT`. Testuje se přes metodu `getForEntity` třídy `RestTemplate`. Testuje se vkládání hlavičky s kontaktními informacemi. Dále se testuje, zda metoda vrací neprázdný textový řetězec pro namátkou zvolené DOI. Jako poslední se testuje ukládání odpovědí Crossref API do repozitáře. Ukládání je nastavené na lokální úložiště. Třída `CrossrefJsonFileRecordingRestTemplate` prochází všemi testy.

Třída `CrossrefSynchronizationServiceImpl` spouští dávkový mód stahování metadat na základě DOI a afiliace autorů k ČVUT. Její testovací třídou je `CrossrefSynchronizationServiceIT`. Ta obsahuje 2 testy, 1. spouští metodu `synchronizeAllCtuRecordsByDoi`, 2. spouští metodu `synchronizeAllCtuRecordsByAffiliation`. Tím se spustí příslušné dávkové módy. Publikační metadata se stahují do vývojářské databáze. Testy se považují za úspěšné, pokud nejsou přerušeny výjimkou a stahování doběhne do konce. Třída testy prochází.

5.3 Testování před nasazením do provozu

Dávkový mód byl před nasazením otestován na testovacím prostředí, kde byl simulován běh dávkového módu stahování metadat na základě DOI. Celkem bylo staženo 17 984 záznamů. Z nich pouze 1 443 (tj. 8 %) záznamů obsahovalo údaje o autorech ČVUT s vyplněnou afiliací k ČVUT.

Pro zjištění počtu výskytů publikací s afiliací autorů k ČVUT byl proveden testovací dotaz Crossref API. Dotaz je uveden v příloze D. Výsledek dotazu ukazuje, že vyhledáváním podle afiliace k ČVUT nelze v době psaní této práce získat více než 350 záznamů.

Testování odhalilo potřebu upravit integritní omezení v databázi systému V3S u atributu `source`. Zatímco v databázích WoS a Scopus je atribut povinný, v Crossref API tomu tak není a v některých záznamech se nevyskytuje, což může způsobit výjimku `DBFunctionException` při ukládání externí publikace do databáze V3S, protože atribut byl v databázi povinný. Problém byl vyřešen odstraněním integritního omezení povinného vyplnění atributu v databázi.

5.3. Testování před nasazením do provozu

V testovacím prostředí běžel dávkový mód stahování podle DOI několik hodin. Za tuto dobu se mohou vyskytnout problémy s dostupností externí služby. Proto jsou případné výjimky zalogovány a pokračuje se zpracováním dalšího záznamu. Díky tomu lze detekovat více chyb při jediném běhu dávkového módu. Dávkový mód vyhledávání podle DOI úspěšně prošel testováním.

Ostrý provoz

Řešení požadavku F3.1 (stahovat metadata publikací ze systému Crossref na základě DOI v dávkovém módu) bylo nasazeno do provozu a poprvé spuštěno 27. 3. 2019. Během prvního běhu bylo staženo 24 855 záznamů. Z nich 1 965 (tj. 8 %) obsahovalo autory s afiliací k ČVUT. V průběhu dalších 33 spuštění do 11. 5. 2019 (včetně) bylo sklizeno dalších 359 publikací. Počet publikací s afiliací některého z autorů k ČVUT se zvýšil o 29. To potvrzuje nízký výskyt těchto dat popsany v sekci 5.3 a dokumentuje, že ani nejnovější přírůstky neindikují významné zvýšení přítomnosti tohoto údaje. Z toho vyplývá, že v současnosti nelze očekávat výrazné množství metadat získaných při běhu dávkového módu stahování metadat ze systému Crossref na základě afiliace autorů (požadavek F3.2).

Závěr

Systém V3S byl rozšířen o modul pro stahování publikačních metadat na základě DOI a afiliací autorů k ČVUT ze systému Crossref. Byla provedena analýza, návrh a implementace celého řešení. Funkčnost dávkové stahování metadat ze systému Crossref na základě DOI byla po otestování uvedena do ostrého provozu dne 27. 3. 2019. Dávkový mód stahování metadat na základě afiliace autorů a interaktivní funkcionalita předvyplnění metadat nového záznamu procházejí v době odevzdání této práce testováním a předpokládá se její nasazení v průběhu měsíce června 2019.

Stanovené cíle práce byly dosaženy.

Literatura

1. *euroCRIS* [online]. euroCRIS. [cit. 2019-05-10]. Dostupné z: <https://www.eurocris.org>.
2. *V3S* [online]. České vysoké učení technické v Praze, 2015. [cit. 2019-4-5]. Dostupné z: <https://v3s.cvut.cz/anonymous/help>.
3. ZÁCHA, P.; ŠPAČEK, J. Kontrola publikací na tabletu i mobilu. *Pražská technika* [online]. 2016, roč. 18, č. 5, str. 31 [cit. 2019-4-10]. ISSN 1213-5348. Dostupné z <https://media.cvut.cz/sites/media/files/content/publications/099ba69c-36d6-4a76-b467-a6b1f07de05d/0951ee4e-a86e-4bcf-8fb6-1c1b2673fd24.pdf>.
4. DVOŘÁK, J. et al. Practical CRIS Interoperability. In: *Procedia Computer Science* [online]. Číslo 146, str. 254-264. ISSN 1877-0509. [cit. 2019-05-10]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S187705091930081X?via%3Dihub#cebibsec1>.
5. *Web of Science* [online]. Clarivate, 2019. [cit. 2019-05-10]. Dostupné z: <https://clarivate.com/products/web-of-science>.
6. *Scopus: Access and use Support Center* [online]. Elsevier, 2019. [cit. 2019-05-10]. Dostupné z: https://service.elsevier.com/app/answers/detail/a_id/15534/supporthub/scopus/#tips.
7. *Digital Object Identifier System* [online]. International DOI Foundation [cit. 2019-4-5]. Dostupné z: <https://www.doi.org/hb.html>.
8. *Crossref* [online]. PUBLISHERS INTERNATIONAL LINKING ASSOCIATION, INC. (2018). [cit. 2019-4-19]. Dostupné z: <https://www.crossref.org/about>.

9. *Cambridge Dictionary* [online]. Cambridge University Press, 2019. [cit. 2019-05-08]. Dostupné z: <https://dictionary.cambridge.org/dictionary/english/affiliation>.
10. Crossref [online]. *Crossref REST API*, 2013 [cit. 2019-4-12]. Dostupné z: <https://github.com/CrossRef/rest-api-doc>.
11. *Liquibase* [online]. Datical. [cit. 2019-05-12]. Dostupné z: <https://www.liquibase.org>.
12. SCHILDT, H. *Java: The Complete Reference*. 8. edice. McGraw-Hill, 2011. ISBN 9780071606301.
13. *Spring* [online]. Pivotal Software, Inc. [cit. 2019-04-20]. Dostupné z: <https://spring.io/projects/spring-framework>.
14. *Extensible Markup Language (XML) 1.0 (Fifth Edition)* [online]. W3C, 2019. [cit. 2019-05-08]. Dostupné z: <https://www.w3.org/TR/REC-xml/#sec-origin-goals>.
15. TYSON, M. What is JSF? Introducing JavaServer Faces. In: *Java-World* [online]. 2018. ISSN 10918906. [cit. 2019-05-08]. Dostupné z: <http://ezproxy.techlib.cz/login?url=https://search-proquest-com.ezproxy.techlib.cz/docview/2173760806?accountid=119841>.
16. LAURENČÍK, M.; BUREŠ, M. *SQL – Podrobný průvodce uživatele*. Praha: GRADA, 2018. 216 stran. ISBN 978-80-271-0774-2.
17. SIVA PRASAD REDDY, K. *Java Persistence with MyBatis 3* [online]. UK: Packt Publishing, 2013. ISBN 978-1-78216-680-1. [cit. 2019-04-30]. Dostupné z: <https://ebookcentral.proquest.com/lib/techlib-ebooks/detail.action?docID=1210003>.
18. *Apache Maven Project* [online]. The Apache Software Foundation, 2019. [cit. 2019-05-12]. Dostupné z: <https://maven.apache.org>.
19. *Jackson Project Home @github* [online]. Github, 2019. [cit. 2019-05-04]. Dostupné z: <https://github.com/FasterXML/jackson>.
20. SMART, F. J. *Jenkins: The Definitive Guide: Continuous Integration for the Masses* [online]. O'Reilly Media, Inc., 2011 [cit. 2019-4-8]. ISBN 144931306X, 9781449313067. dostupné z: <https://books.google.cz/books?id=4bjDCQAAQBAJ&lpg=PR3&ots=FRhprwHv-Q&dq=%20jenkins%20%22continuous%20integration%22%20-box&lr&pg=PA7#v=onepage&q&f=false>.

21. BARBAGLIA, G.; MURZILLI, S.; CUDINI, S. Definition of REST web services with JSON schema. In: *Software: Practice and Experience* [online]. Ročník 46, číslo 6, str. 907-920. ISSN 1097-024X. [cit. 2019-04-20]. Dostupné z: <https://onlinelibrary-wiley-com.ezproxy.techlib.cz/doi/full/10.1002/spe.2466>.
22. BRAY, T. RFC 8259. *The JavaScript Object Notation (JSON) Data Interchange Format*. ISSN 2070-1721. Internet Engineering Task Force, 2017-12. [cit. 2019-05-11]. Dostupné z: <https://tools.ietf.org/html/rfc8259>.
23. CHACON, S.; STRAUB, B. *Pro Git* [online]. 2. vydání. APress, 2014. ISBN 978-1484200773. [cit. 2019-05-11]. Dostupné z: <https://git-scm.com/book/en/v2>.
24. *JUnit* [online]. JUnit, 2018. [cit. 2019-05-02]. Dostupné z: <https://junit.org/junit4/>.

Seznam použitých zkratk

ČVUT České Vysoké Učení Technické

WoS Web of Science

DOI Digital Object Identifier item[API] Application Programming Interface

GUI Graphical user interface

XML Extensible markup language

SQL Structured Query Language

JSF Java Server Faces

REST REpresentational State Transfer

JSON JavaScript Object Notation

ISBN International Standard Book Number

ISSN International Standard Serial Number

Obsah přiloženého CD

	readme.txt	stručný popis obsahu CD
	src		
		implukázka zdrojového kódu implementace
		thesiszdrojová forma práce ve formátu \LaTeX
		text text práce
		thesis.pdftext práce ve formátu PDF

Tabulky mapování Crossref JSON na V3S ExtPublication

C. TABULKY MAPOVÁNÍ CROSSREF JSON NA V3S EXTPUBLICATION

Tabulka C.1: Mapování

Typ	Atribut JSON	Atribut(y) Ext-Publication	Typ
String	publisher	publisher	String
String	abstract	abstractOrig	String
String	DOI	doiCode	String
String	issue	issue	String
String	volume	volume	String
String	article-number	articleNumber	String
String	publisher-location	publisherLocation	String
String	page	startPage, endPage, numberOfPages	String
PartialDate	indexed	whenAddedToIndex	String
PartialDate	issued	issueDate, year	String, Integer
Array<String>	title, original-title, short-title, sub- title	title	String
String	container-title, short-container- title	source	String
Array<String>, Array<ISSN>	ISSN, issn-type	issn, eIssn	String
Array<String>, Array<ISSN>	ISBN, isbn-type	isbn, eIsbn	String
Array <Contributor>	author	authors	List <ExtAuthors>
Array<Resource>	link	links	Array<String>

Tabulka C.2: Mapování Autorů

Typ	Atribut Contri- butor	Atribut(y) Ext- Publication	Typ
String	family	surname, initials	String
String	given	name, initials	String
URL	ORCID	orcid	String
Array <Affiliation>	affiliation	affiliations	List <ExtAffiliation>

Tabulka C.3: Mapování typů

Typ Crossref API	Typ systému V3S
journal-article	DocType.ARTICLE
book-part	DocType.BOOK_CHAPTER
book	DocType.BOOK
book-set	SourceType.BOOK_SERIES
proceedings-article	DocType.PROCEEDINGS_PAPER
journal	SourceType.JOURNAL
book-chapter	DocType.BOOK_CHAPTER
proceedings	SourceType.CONFERENCE_PROCEED
reference-book	DocType.BOOK

Dotaz na afiliace ČVUT v Crossref API

```
https://api.crossref.org/works?query.affiliation=CTU&query.affiliation=
CVUT&query.affiliation=C%CC%8CVUT&query.affiliation=C%CC%8Ceske%CC%
81vysoke%CC%81uc%CC%8Ceni%CC%81technicke%CC%81&query.affiliation=
Czechtechnicaluniversity
```