Návrh systému

Nikola Hornikova, Filip Kerak, Iveta Balintova, Erik Bily

TIS 2019

## Úvod

### 1.1. Účel dokumentu

Tento dokument predstavuje podrobný popis návrhu pre systém na konverziu ohlasov. Pomocou diagramov a detailných popisov v tomto dokumente je vysvetlené, akým spôsobom bude systém vyvinutý a ako bude následne fungovať tak, aby splnil všetky požiadavky uvedené v katalógu požiadaviek.

### 1.2. Zameranie a rozsah návrhu

Predpokladáme, že čitateľ tohto dokumentu má dôkladne prečítaný katalóg požiadaviek a teda má ucelenú predstavu ako má softvér fungovať. V časti návrh implementácie sú popísané všetky triedy z triedneho diagramu. Veľkú časť tohto dokumentu tvoria UML diagramy, v ktorých je dôkladne predstavené, na aké moduly a triedy bude systém rozdelený. V časti UML diagramy sú predstavené všetky diagramy na základe, ktorých je dobre vysvetlený priebeh celého systému.

### 1.3. Slovník pojmov

CREPČ2 - Centrálny register evidencie publikačnej činnosti SR ([http://app.crepc.sk](http://app.crepc.sk/))

Záznam o publikácií - súhrn bibliografických údajov/metadát jednoznačne identifikujúci publikáciu

Ohlas - informácia o publikácií, ktorá cituje, respektíve recenzuje inú publikáciu (ak autor cituje sám seba, ide o autocitáciu, ktorá sa do KIS neeviduje)

XML - rozšíriteľný značkovací jazyk, umožňuje jednoduché vytváranie konkrétnych

značkovacích jazykov na rôzne účely a široké spektrum rôznych typov údajov

MARC21 - bibliografický výmenný formát

Knižnično-informačný systém (Virtua) - komplexný knižničný systém používaný

univerzitou

## Obsluha aplikácie

### 2.1. Konzola

Usage: python app1\_initializer -o DDMMYYYY [-h | --help] [-d <date> | --do=<date> ]

[--out-file=<file>] [--out-path=<path>]

[--log-file=<file>] [--log-path=<path>]

Povinny argument:

-o <date>, --od=<date> Datum od ktoreho ziskava ohlasy, date musi byt vo formate DDMMYYYY

Volitelne argumenty:

-d <date>, --do=<date> Datum po ktory ziskava ohlasy, date musi byt vo formate DDMMYYYY, default je dnesny datum

--out-file=<file> Nazov suboru do ktoreho sa uklada vystup,

default nazov je &'output'. V pripade ze subor neexistuje, vytvori sa novy subor s tymto menom

--out-path=<path> Cesta k vystupnemu suboru. &relativna / absolutna ,

default je ./output/

--log-file=<file> Nazov suboru do ktoreho sa vypisuje priebeh aplikacie,

default nazov je &'log'. V pripade ze subor neexistuje, vytvori sa novy

--log-path=<path> Cesta k log suboru. &relativna / absolutna ?,

default je ./output/

## Používané technológie

### 3.1. Python

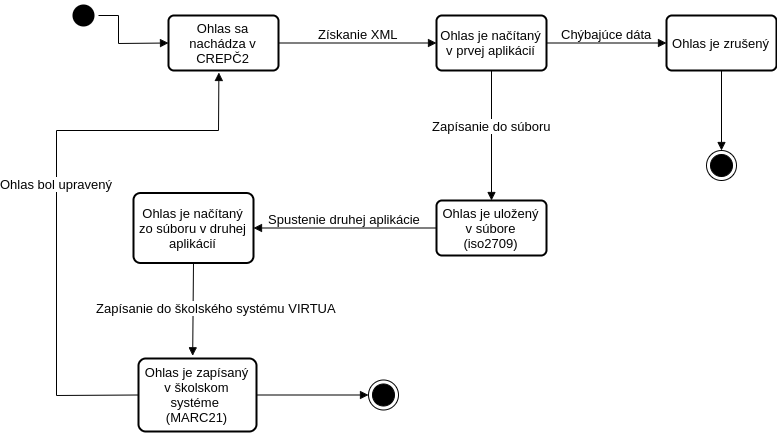
V tomto interpretovanom, interaktívnom programovacom jazyku budú napísané obe aplikácie z ktorých sa navrhnutý systém skladá.

### 3.2. Terminál

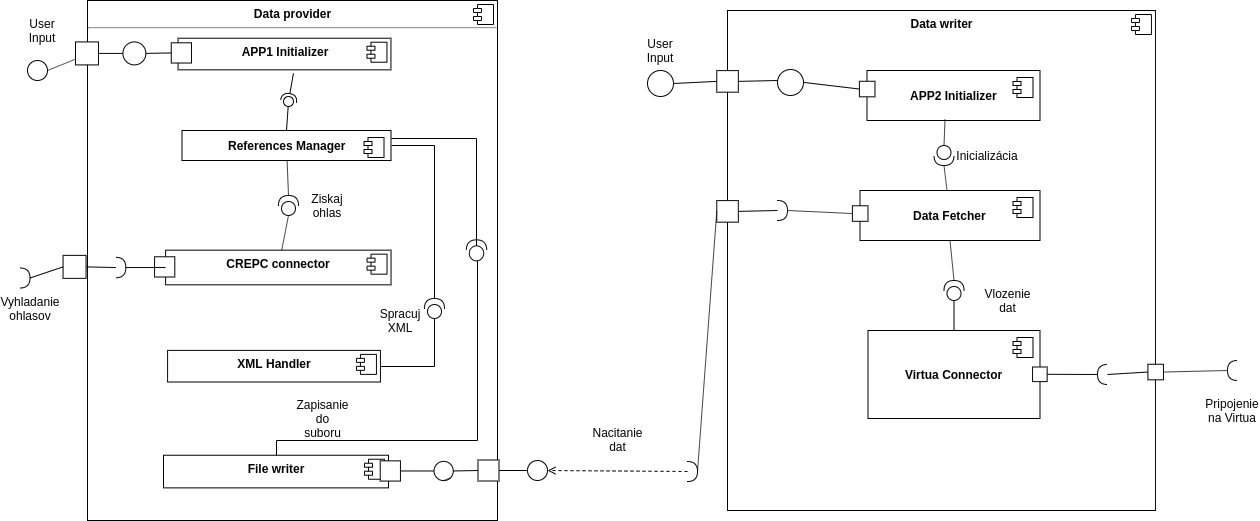
Obe aplikácie systému budú ovládané z konzoly, podľa návodu v kapitole 2 časti 2.1.

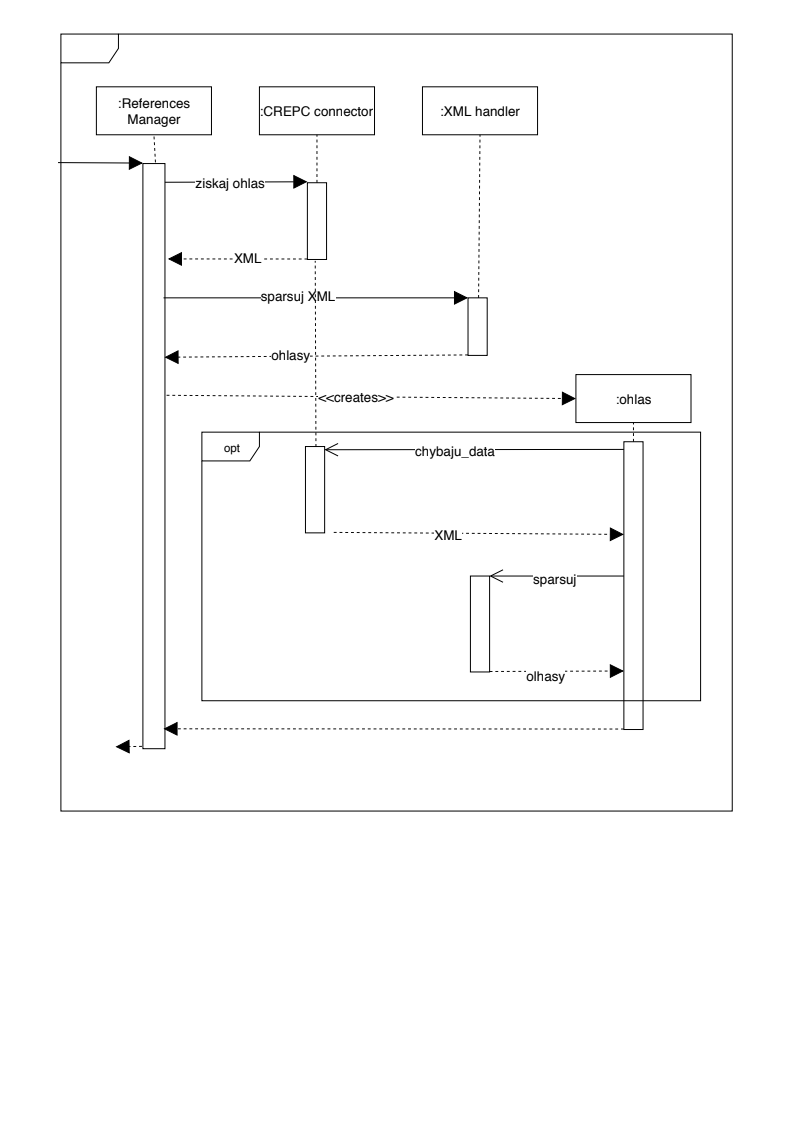
## UML Diagramy

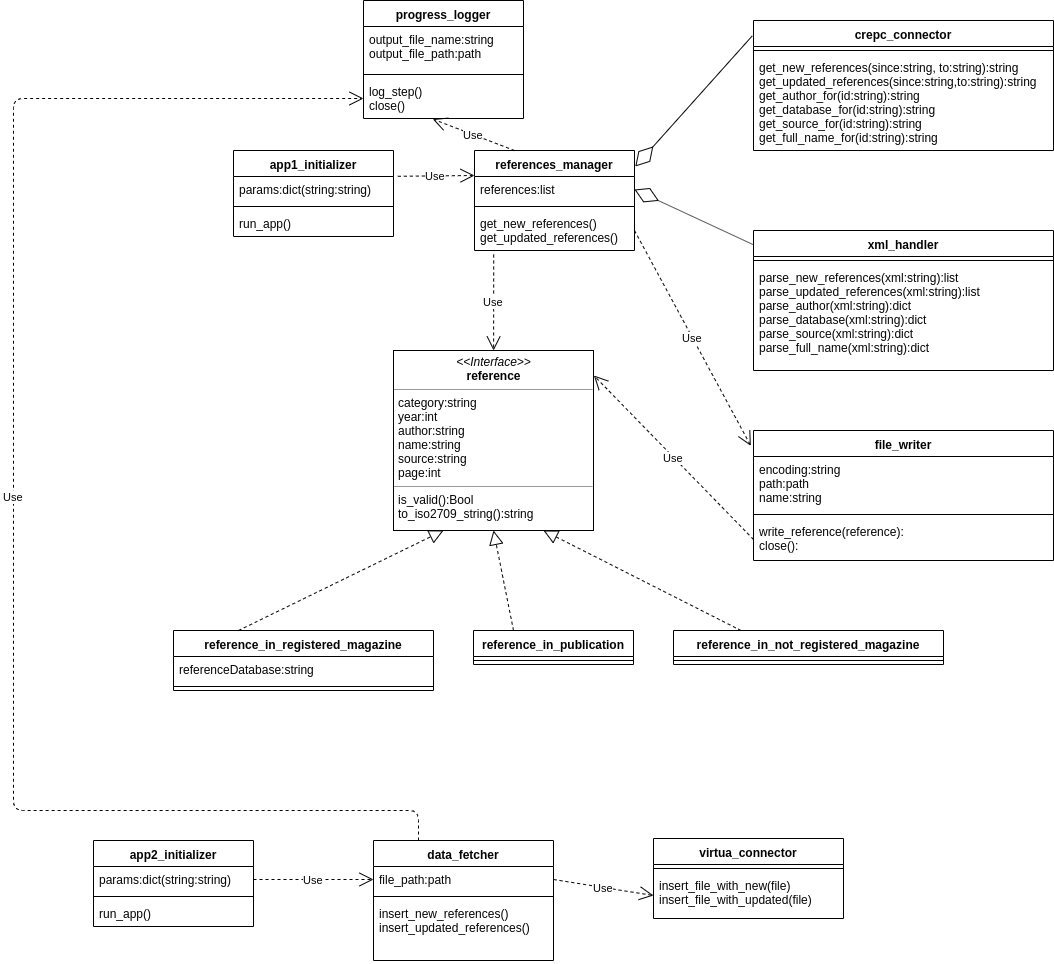
* 1. UML State diagram



* 1. UML Component diagram



* 1. UML Sequence diagram
  2. UML Class diagram



## Návrh implementácie

Systém je rozdelený do dvoch aplikácií. Nižšie sú uvedené všetky triedy a

metódy, ktoré bude potrebné implementovať na základe class diagramu z kapitoly 4.4 UML Class diagram. Sú oddelené na základe toho, ktorá aplikácia ich využíva.

* 1. Aplikácia 1

Prvá aplikácia sa stará o získanie dát a ich následné zapísanie do súboru v

správnom formáte.

* + 1. Trieda app1\_initializer

Funkciou tejto triedy je zabezpečenie inicializácie a následné spustenie aplikácie 1.

Metódy:

* \_\_init\_\_(self, params)
  + Ako parameter dostane argumenty zadané pri spustení, tie spracuje a uloží do vnútornej reprezentácie.
  + Pri nezadaní alebo nesprávnom zadaní argumentov vypíše možnosti spustenia.
  + params - argumenty pri spustení (typ: list[string])
  + triedne premenné: file\_path, since, to, log\_path
* run\_app(self)
  + Vyrobí inštanciu progress\_logger a file\_writer, následne aj inštanciu references\_manager, na ktorej zavolá požadovanú metódu.

* + 1. Trieda file\_writer

Funkciou tejto triedy je zapisovanie ohlasov do súboru.

Metódy:

* \_\_init\_\_(self, name, encoding = “utf-8”, path=””)
  + Pri inicializácií sa pripraví súbor na zapisovanie.
  + name - názov súboru (typ: string)
  + encoding - kódovanie súboru (typ: string, default: “utf-8”)
  + path - cesta, kde bude súbor uložený (typ: string, default: ””)
* write\_reference(self, reference)
  + Zapíše do súboru jeden ohlas vo forme MARC21/iso2709
  + reference - ohlas na zapisanie (typ: reference)
* close(self)
  + Ukončí zápis a zavrie súbor.
    1. Trieda progress\_logger

Trieda zabezpečuje vypisovanie priebehu:

1. Do konzoly postupne po krokoch.
2. Do súboru po zavolaní close.

Metódy:

* \_\_init\_\_(self, output\_file\_name, output\_file\_path=””)
  + Pri inicializácií dostane ako parametre meno a umiestnenie výsledného súboru.
  + Vyrobí tento súbor a uloží si ho pre budúci zápis.
  + output\_file\_name - meno výsledného súboru (typ: string)
  + output\_file\_path - cesta, kam sa má zapísať výsledný súbor (typ: string, default: “”)
* log\_step(self, reference)
  + Zapíše správu o úspešnom zapísaní ohlasu.
  + reference - ohlas na zapísanie (typ: reference)
* log\_error(self, error)
  + Zapíše správu o chybe pri spracovaní ohlasu.
  + error - vzniknutá chyba (typ: exception)
* close(self)
  + Zapíše všetky zostávajúce zmeny do súboru a zavrie ho.
    1. Trieda reference

Trieda uchováva jeden ohlas. Je abstraktná, všetky jej metódy sú abstraktné a odvodené sú od nej tri podtriedy:

1. reference\_in\_not\_registered\_magazine

* Uchováva ohlas z neregistrovaného časopisu.

1. reference\_in\_publication

* Uchováva ohlas z publikácie.

1. reference\_in\_registered\_magazine

* Uchováva ohlas z registrovaného časopisu.

Metódy:

* \_\_init\_\_(self, data)
  + Nastaví si hodnoty z parametra data.
  + Chýbajúce hodnoty sa pokúsi získať.
  + data - slovník hodnôt pre vytvorenie ohlasu, spresňuje odvodená trieda (typ: dictionary{string:string})

* is\_valid(self)
  + vracia bool hodnotu - vráti True, ak je ohlas kompletný a je možné ho zapísať
* to\_iso2709\_string(self)
  + vracia string - reťazec reprezentujúci ohlas v tvare MARC21/iso2709

* \_\_str\_\_(self)
  + vracia string - reťazec reprezentujúci ohlas v zrozumiteľnom tvare
    1. Trieda reference\_manager

Trieda poskytuje možnosť na získanie ohlasov.

Metódy:

* \_\_init\_\_(self, initializer, logger, writer)
  + initializer - obsahuje argumenty zo spustenia (typ: app1\_initializer)
  + logger - logger do ktorého sa bude zapisovať priebeh (typ: progress\_logger)
  + writer - writer, do ktorého sa zapíšu ohlasy (typ: file\_writer)
* get\_new\_references(self)
  + do writeru zadanom pri inicializácií zapíše novovzniknuté ohlasy, pričom sa použijú argumenty z initializer
* get\_updated\_references(self)
  + do writeru zadanom pri inicializácií sa zapíše upravené ohlasy, pričom sa požijú argumenty z initializer
    1. Trieda crepc\_connector

Trieda zabezpečuje komunikáciu s centrálnym registrom publikačnej činnosti.

Metódy:

* \_\_init\_\_(self)
* get\_new\_references(self, since, to=””)
  + Získa novovzniknuté ohlasy.
  + since - dátum od ktorého sa budú získavať ohlasy (typ: string)
  + to - dátum po ktorý sa budú získavať ohlasy (typ: string, default: “”)
  + vracia string - reťazec obsahujúci XML so zoznamom ohlasov
* get\_updated\_references(self, since, to=””)
  + Získa upravené ohlasy.
  + since - dátum od ktorého sa budú získavať ohlasy (typ: string)
  + to - dátum po ktorý sa budú získavať ohlasy (typ: string, default: “”)
  + vracia string - reťazec obsahujúci XML so zoznamom ohlasov
* get\_author\_for(self, id)
  + Získa celé meno autora pre zadané id.
  + id - id autora (typ: string)
  + vracia string - celé meno autora
* get\_database\_for(self, id)
  + Získa názov databázy pre zadané id.
  + id - id databázy (typ: string)
  + vracia string - názov databázy
* get\_source\_for(self, id)
  + Získa celý názov zdroja pre zadané id.
  + id - id zdroja (typ: string)
  + vracia string - celý názov zdroja
* get\_full\_name\_for(self, id)
  + Získa celý názov publikácie pre zadané id.
  + id - id publikácie (typ: string)
  + vracia string - celý názov publikácie
    1. Trieda xml\_handler

Trieda poskytuje funkcie na spracovanie rôznych xml vstupov.

Metódy:

* \_\_init\_\_(self)
* parse\_new\_references(self,xml)
  + Spracuje novovzniknuté ohlasy.
  + xml - reťazec s XML na spracovanie (typ: string)
  + vracia list[dictionary{string:string}] - zoznam slovníkov, jeden slovník reprezentuje dáta z jedného ohlasu
* parse\_updated\_references(self, xml)
  + Spracuje aktualizované ohlasy.
  + xml - reťazec s XML na spracovanie (typ: string)
  + vracia list[dictionary{string:string}] - zoznam slovníkov, jeden slovník reprezentuje dáta z jedného ohlasu
* parse\_author(self, xml)
  + Spracuje xml obsahujúce meno autora.
  + xml - reťazec s XML na spracovanie (typ: string)
  + vracia string - meno autora
* parse\_database(self, xml)
  + Spracuje xml obsahujúce názov databázy.
  + xml - reťazec s XML na spracovanie (typ: string)
  + vracia string - názov databázy
* parse\_source(self, xml)
  + Spracuje xml obsahujúce zdroj ohlasu.
  + xml - reťazec s XML na spracovanie (typ: string)
  + vracia string - názov zdroja
* parse\_full\_name(self, xml)
  + Spracuje xml obsahujúce celý názov publikácie.
  + xml - reťazec s XML na spracovanie (typ: string)
  + vracia string - názov publikácie
  1. Aplikácia 2

Druhá aplikácia sa stará o vloženie pripravených údajov do MARC21

prostredníctvom nástrojov univerzitného knižničného systému.

* + 1. Trieda app2\_initializer

Funkciou tejto triedy je zabezpečenie inicializácie a následné spustenie aplikácie 2.

Metódy:

* \_\_init\_\_(self, params)
  + Ako parameter dostane argumenty zadané pri spustení, tie spracuje a uloží ich do vnútornej reprezentácie.
  + Pri nezadaní alebo nesprávnom zadaní argumentov vypíše možnosti spustenia.
  + params - argumenty pri spustení (typ: list[string])
* run\_app(self)
  + Vyrobí inštanciu data\_fetcher a následne spustí vkladanie údajov
    1. Trieda data\_fetcher

Pracuje s triedou virtua\_connector, ktorej podáva informácie a vkladá ich do systému.

Metódy:

* \_\_init\_\_(self, file\_path)
  + file\_path - cesta k súboru s dátami na vloženie (typ: string)
* insert\_new\_references(self)
  + Začne vkladanie novovzniknutých ohlasov.
* insert\_updated\_references(self)
  + Začne vkladanie upravených ohlasov.
    1. Trieda virtua\_connector

Funkciou tejto triedy je zabezpečiť komunikáciu s knižnično-informačným systémom Virtua.

Metódy:

* \_\_init\_\_(self)
* insert\_file\_with\_new(self, file)
  + Vloží súbor s novovzniknutými ohlasmi.
  + file - súbor na vloženie
* insert\_file\_with\_updated(self, file)
  + Vloží súbor s upravenými ohlasmi.
  + file - súbor na vloženie

### 