

Knihovna PyWSDP pro přístup k webovým službám dálkového přístupu do KN

Ing. Linda Kladivová ^a

^a ČVUT v Praze, Fakulta stavební, Katedra geomatiky, Thákurova 7, 166 29, Praha 6

ABSTRAKT

Webové služby dálkového přístupu (WSDP) umožňují registrovaným uživatelům získávat údaje z Informačního systému katastru nemovitostí (ISKN) v rámci svého vlastního programového rozhraní. Knihovna PyWSDP je open-source projekt vyvíjen v jazyce Python, který zpřístupňuje a zjednodušuje práci s WSDP. Nabízí intuitivní rozhraní s možností čtení a zápisu dat v několika datových formátech. Součástí knihovny je dokumentace obohacená o interaktivní příklady použití z prostředí Jupyter Notebooks.

KLÍČOVÁ SLOVA

PyWSDP, WSDP, KN, SPI, SGI, SOAP, WSDL, ČtiOS

ÚVOD

PyWSDP je open-source knihovna nabízející rozhraní pro přístup k Webovým službám dálkového přístupu do Katastru nemovitostí (dále jen WSDP), které jsou programovou Simple Object Access Protocol (SOAP) variantou aplikace Dálkový přístup do katastru nemovitostí (DP). Knihovna ve verzi 2.0 obsahuje dva hlavní moduly: ČtiOS a GenerujCenoveUdajeDleKu. Ty zjednodušují práci se dvěma WSDP službami – samostatně stojící službou ČtiOS a službou *Generování cenových údajů podle katastrálního území*, která je součástí většího celku s názvem *Sestavy*. Pro tyto služby moduly nabízejí intuitivní rozhraní pro dotazování a čtení/zápis dat v několika datových formátech. Objektový návrh knihovny je díky architektuře postavené na několika návrhových vzorech snadno rozšiřitelný o další moduly. Ke knihovně byla sepsána podrobná dokumentace, která obsahuje zejména popis veřejného programového rozhraní knihovny a konkrétní příklady použití modulů zpracované pomocí webového nástroje Jupyter Notebooks.

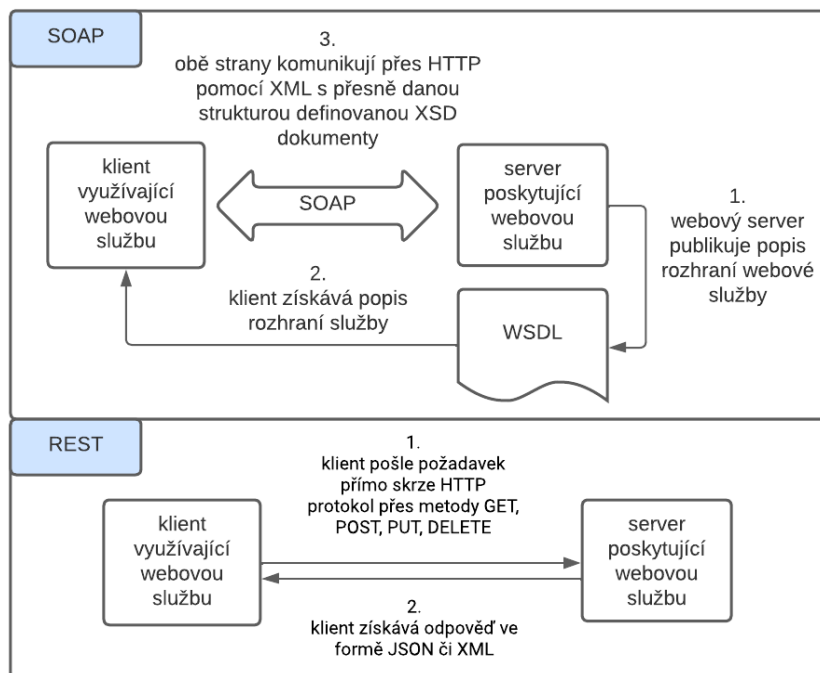
Historie a motivace

Tvorba knihovny započala v roce 2018 v rámci předmětu 155FGIS na Fakultě stavební ČVUT v Praze s cílem vytvořit objektový návrh v jazyce Python zakomponováním několika návrhových vzorů. Vývoj knihovny probíhal s přestávkami, nicméně stálý zájem veřejnosti o programové řešení WSDP požadavků ukazoval, že v open-source světě stále neexistuje komplexní řešení ani pro deanonymizaci oprávněných subjektů (existují pouze komerční řešení např. v softwaru MYSIS [1]) ani pro práci s dalšími službami DP poskytovanými ČÚZK.

Zásadní motivací pro opětovný vývoj a dotažení knihovny do verze 2.0 bylo její plánované využití pro disertační práci autorky, jejíž téma se týká vytvoření vektorových dlaždic, dynamické aktualizace vektorových dlaždic a operací nad dlaždicemi v souvislosti s daty katastru nemovitostí (KN). PyWSDP zde poslouží jako příklad SOAP klienta, který se pokusí využít atributové povahy vektorových dlaždic. To může vést k potenciální možnosti propojení souboru geodetických informací a souboru popisných informací v koncové mapové aplikaci.

Webové služby dálkového přístupu

Aplikace WSDP je postavena na architektonickém stylu SOAP/WSDL webové služby. SOAP popisuje a implementuje způsob komunikace přes Hypertext Transfer Protocol (HTTP) mezi webovým serverem a klientem, k čemuž využívá XML soubory (zkratka pro Extensible Markup Language). Oproti druhému využívanému Representational State Transfer architektonického návrhu [2] (zkráceně REST), který zprostředkovává komunikaci přímo skrze HTTP protokol přes metody GET, POST, PUT, DELETE, představuje SOAP robustnější návrh. XML požadavky a odpovědi serveru jsou strukturované na základě silných pravidel, které vychází z přidružených souborů obsahujících definice XML schémat (soubory známé pod zkratkou XSD). Aby klient mohl jednoduše přistupovat k definovanému rozhraní, je dále SOAP rozšířeno o WSDL dokumenty, které pomocí jazyka Web Services Description Language (WSDL) popisují webové služby jako takové - přístupový bod, metody, názvy vstupních a výstupních parametrů a jejich datové typy [3]. Diagramy obou architektur jsou znázorněny na Obrázku 1.



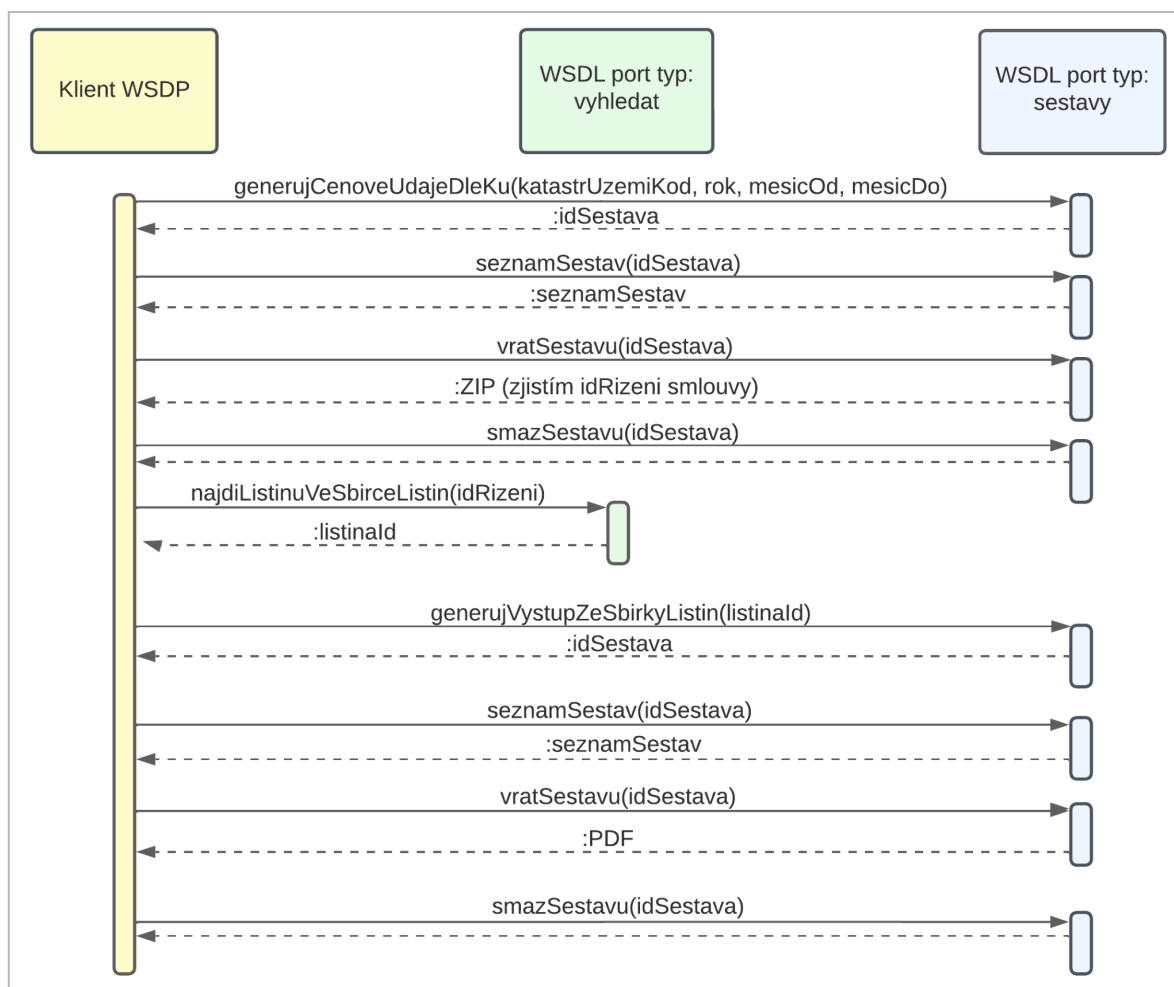
Obrázek 1. SOAP versus REST architektura webových služeb (upraveno na základě [6] a [7])

Popisy WSDP služeb jsou obsaženy v šesti WSDL souborech odpovídajícím pěti skupinám nabízených služeb - *Sestavy*, *Číselníky*, *Informace*, *Správa účtu* a *Vyhledávání* a samostatně stojící službě *ČtiOS*. Každý soubor WSDL se vztahuje k jednomu přístupovému bodu a definuje metody (naše WSDP služby), které mohou být nad určitým přístupovým bodem

volány. To koresponduje s tím, že SOAP je procedurálně orientovaný architektonický styl webových služeb, zatímco REST je orientován datově [4].

První skupina služeb *Číselníky* je brána jako podpůrná a má 33 metod, které slouží k hledání informací v číselnících KN (např. vypiš seznam obcí začínající na „Poz“). Druhá skupina služeb *Informace* je rovněž podpůrného charakteru a obsahuje 3 metody, které vrátí XML dokumenty s informacemi podle zadání. Hlavní skupinu webových služeb tvoří *Sestavy*, které obsahují 3 metody pro správu sestav (seznam, zaúčtování a smazání) a 19 metod pro vytváření sestav. Výstupy sestav mohou být buď ve formátu XML nebo mohou být zakódovány jako soubory v odpovědi při zaúčtování sestavy ve formátech PDF, HTML nebo ZIP (pouze pro cenové údaje dle katastrálního území). Často navazující službou na *Sestavy* je některá ze 7 metod ze skupiny *Vyhledávání*, které podle zadání například mohou podle zadaného ID řízení najít listinu ve sbírce listin.

Jeden z příkladů často využívaného propojení více služeb je znázorněn na Obrázku 2. V souborech cenových údajů získaných ze služby *generujCenoveUdajeDleKu* zjistíme jednotky, budovy a parcely, které mají cenový záznam a k nim příslušnou cenu a číslo řízení, kterým se cena zapsala do katastru nemovitostí. Díky číslu řízení pak můžeme skrze službu *najdiListinuVeSbirceListin* najít identifikátor příslušné kupní smlouvy. Pro vygenerování PDF souboru této smlouvy zavoláme na závěr službu *generujVystupZeSbirkyListin*.



Obrázek 2. Příklad řetězení WSDP služeb

Skupiny služeb WSDP jsou aktuálně ve verzi 2.9, služba ČtiOS stojící samostatně od těchto skupin je ve verzi 2.8. Všechny služby mají definované chyby, ke kterým může dojít například při špatně zadaných vstupních parametrech. Významy chybových kódů a další specifika služeb jsou nad rámec tohoto příspěvku a jsou k dispozici ve WSDP dokumentaci [5].

METODIKA

Knihovna je vyvíjena s cílem využití zejména v automatizačních skriptech, proto byl jako programovací jazyk zvolen Python. Původně byly XML požadavky sestavovány za běhu podle připravené šablony, od tohoto postupu bylo ve verzi 2.0 upuštěno. Nyní jsou požadavky sestavovány na základě WSDL souboru daného přístupového bodu pomocí knihovny Zeep¹. Vývoj PyWSDP probíhá na webové službě GitHub², součástí projektu jsou automatizované akce na vytváření dokumentace pomocí knihovny Sphinx s ukázkovými příklady v prostředí Jupyter Notebooks (viz Obrázek 3).

Načteme identifikátory z databáze (pro účely ukázky omezíme počet načtených identifikátorů na 5).

```
[4]: parametry_cti0S_db = ctios.nacti_identifikatory_z_db(db_path, "SELECT * FROM OPSUB
```

Pošleme požadavek a vypíšeme si statistiku zpracovaných identifikátorů.

```
[5]: slovník, slovník_chybnych = ctios.posli_pozadavek(parametry_cti0S_db)
```

```
cti0S      - INFO      -  
cti0S      - INFO      - Požadovaná akce byla úspěšně provedena.  
cti0S      - INFO      - POSIDENT /A0rBG0FR+4GtqsAv0XC/3kKd00CEPBVRfEsIPHeW3CoGcU  
cti0S      - INFO      - POSIDENT +lQ0Ne+Jhd5K4uvds1/KT8qqJcoA/eJ50loIoqvCQi04X93  
cti0S      - INFO      - POSIDENT +VKfYaKhJDrRHUWeS2e7x07g+HhFQapZMPa5kamaF/wS0s6  
cti0S      - INFO      - POSIDENT /ipU68hnrq1J7ZDi4JBaAyBiKjxZzpGZSVIfv8yFMKSKy3  
cti0S      - INFO      - POSIDENT 00ep7V5fFBZoFMe0fcAOPMRtu0G9svLcsDyC/fYkjk3EbFI  
cti0S      - INFO      - Celkový počet dotazovaných identifikátorů na vstupu: 5  
cti0S      - INFO      - Počet odstraněných duplicitních identifikátorů: 0  
cti0S      - INFO      - Počet požadavků, do kterých byl dotaz rozdělen (počet do  
cti0S      - INFO      - Realný úspěšně zpracovaných identifikátorů: 5  
cti0S      - INFO      - Počet neplatných identifikátorů: 0  
cti0S      - INFO      - Počet expirovaných identifikátorů: 0  
cti0S      - INFO      - Počet identifikátorů k neexistujícím OS: 0
```

Uložíme získané osobní údaje do databáze.

```
[6]: vystup, vystup_chybnych = ctios.uloz_vystup(  
    slovník, vystupni_adresar, OutputFormat.GdalDb, slovník_chybnych  
)
```

```
cti0S      - INFO      - Vystup byl uložen zde: /home/linduska/pywspd/tests/data/
```

Obrázek 3. Ukázka z dokumentace - výukový materiál pro modul ČtiOS v prostředí Jupyter Notebooks

¹ <https://docs.python-zeep.org/en/master/>

² <https://github.com/ctu-geoforall-lab/pywspd>

Moduly knihovny

Knihovna PyWSDP podporuje ve své verzi 2.0 práci se samostatně stojící službou ČtiOS a službou *Generování cenových údajů podle katastrálního území*, která je součástí většího celku s názvem Sestavy. Každá z hlavních služeb celku Sestavy (ve verzi 2.0 zatím pouze jedna) zároveň podporuje SOAP služby na správu sestav jako je vypsání informací o sestavě, zaúčtování sestavy a smazání sestavy. Vzhledem k tomu, že veřejné rozhraní knihovny nabízí kromě dotazování WSDP služeb další funkcionality jako například uložení výstupu do různých formátů, mluvíme o modulech.

Ve výměnném formátu katastru (VFK) stále najdeme data o objektech katastru nemovitostí,, ale popisná data s osobními údaji již zde z důvodu nařízení GDPR vyplněna nejsou. Místo ID oprávněného subjektu je vyplněn tzv. pseudoidentifikátor, který slouží k deanonymizaci oprávněných subjektů prostřednictvím webové služby ČtiOS. Modul *CtiOS* potom poskytuje rozhraní, které se kromě komunikace se službou ČtiOS stará o sestavování dotazů na základě vstupních parametrů v datových formátech SQLite databáze a JSON souboru. Podobně nabízí uložení výsledku ve formě SQLite databáze, JSON souboru a také CSV souboru.

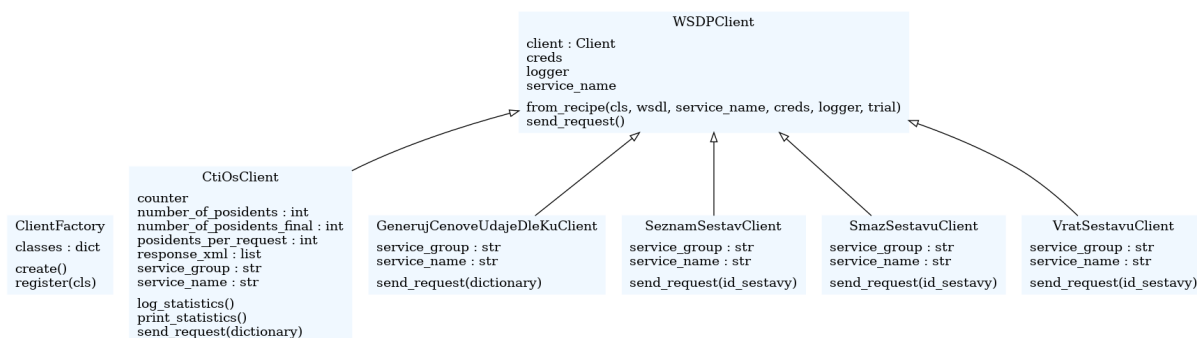
Modul *GenerujCenoveUdajeDleKu* zprostředkovává generování údajů o dosažených cenách nemovitostí podle zadaného katastrálního území pro dané časové období. Po dotazu na server je vytvořena sestava, se kterou je poté možné přes rozhraní dále pracovat - vypsát údaje o sestavě (cena, stav sestavy apod.), zaúčtovat sestavu a smazat sestavu. Rovněž získaný soubor sestavy můžeme dekódovat do formátu ZIP obsahujícího jednotlivé XML soubory cenových údajů pro daná časová období. Správcovské moduly sestav (jako je např. zaúčtování sestavy) jsou při známém ID sestavy rovněž použitelné i samostatně.

Architektura knihovny

Objektový návrh knihovny využívá vhodné návrhové vzory ze skupiny GOF [8] a zároveň je ovlivněn specifiky jazyka Python. Základním použitým návrhovým vzorem je Factory nebo-li tovární metoda, návrhový vzor zařazený ve skupině Creational patterns. Tento vzor je vhodný zejména při vytváření instancí velmi podobných objektů, které v našem případě představují PyWSDP klienti. Ti jsou sice v mnoha ohledech stejní, ale pracují s odlišnými SOAP metodami. Při návrhu Factory metody bylo využito dynamického registrování tříd v Pythonu pomocí dekorátorů³, kdy nejprve třídy registrujeme (vytvoříme instance jednotlivých klientů), a až dodatečně se podle vstupních parametrů rozhodneme, jakou z registrovaných tříd inicializujeme.

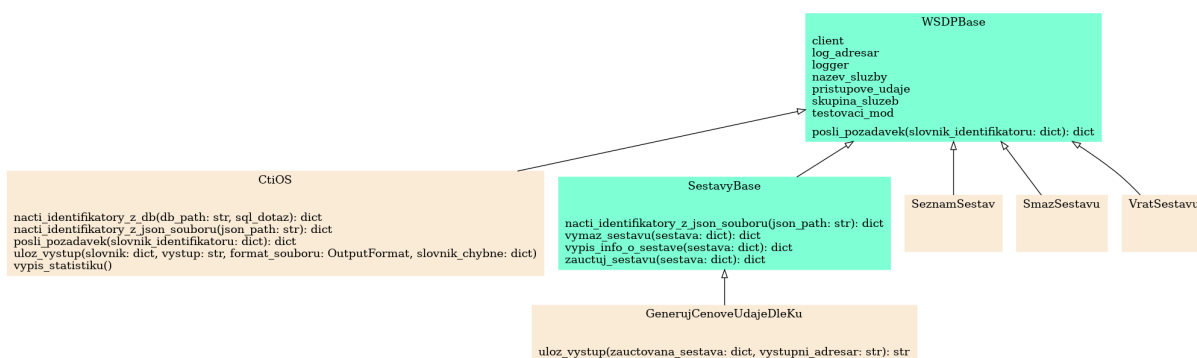
Jednotliví klienti jsou součástí interního API knihovny a musí implementovat metodu *send_request*. Pro tento účel a také pro účely inicializace mají svého předka reprezentovaného abstraktní třídou *WSDPClient*. Tato architektura odráží návrhový vzor Template patřící do skupiny Behavioral patterns, který se využívá při implementování metod se stejnou strukturou, ale s odlišným chováním. Architektura interního API knihovny je zakreslena v UML diagramu na Obrázku 4.

³<https://medium.com/@geoffreykoh/implementing-the-factory-pattern-via-dynamic-registry-and-python-decorators-479fc1537bbe>



Obrázek 4. UML diagram interního API PyWSDP v2.0 přístupujícího k samotným WSDP službám pomocí zeep knihovny

Obsahem knihovny je rovněž několik dalších pomocných tříd, které se starají o zpracování dat. Příkladem může být třída DbManager zajišťující práci s SQLite databází pro modul ČtiOS. Logika klientských a pomocných tříd je dále sdružena do samostatných systémů majících jednotné rozhraní pro funkcionalitu, kterou systém poskytuje. Tento systém patří do Structural patterns a nazývá se Facade (česky fasáda). Jednotlivé třídy Fasády jsou v našem případě pojmenovány moduly. Ty tvoří společně s rodičovskými třídami SestavyBase a WSDPBase veřejné API knihovny (viz Obrázek 5).



Obrázek 5. UML diagram veřejného API PyWSDP v2.0 (základní třídy jsou vyznačeny zeleně)

ZÁVĚR

I přes neustálý zájem veřejnosti dlouho neexistovalo v open-source světě žádné řešení zpřístupňující webové služby dálkového přístupu do KN. Ač knihovna PyWSDP ve verzi 2.0. je stále daleko od podpory všech nabízených WSDP služeb, byla architektonicky postavena tak, aby byla jednoduše rozšiřitelná. K tomuto účelu byl vymyšlen návrh kombinující tři návrhové vzory - Factory (továrna), Template (šablona) a Facade (fasáda). Knihovna bude dále rozšiřována, aby mohla být využita v rámci disertační práce autorky jako příklad webové SOAP služby volané nad vektorovými dlaždicemi. Pro tento účel bude vytvořena webová aplikace ve vektorových dlaždicích propojující prostorová data katastru (SGI) s popisným operátem (SPI).

PODĚKOVÁNÍ

Děkuji mému školiteli Martinu Landovi, že měl trpělivost s poněkud krkolomnou cestou k finálnímu objektovému návrhu knihovny. Původní návrh se z důvodu mého rozhodnutí o

zapojení knihovny Zeep musel ve verzi knihovny 2.0 lehce pozměnit, což vedlo ke změnám ve veřejném API knihovny.

REFERENCE

- [1] Misys - přehled změn. GEPRO [online] [cit. 31.10.2022] Dostupné z: <https://www.gepro.cz/produkty/misys/misys-prehled-zmen/>
- [2] Tihomirovs, J., & Grabis, J. (2016). Comparison of SOAP and REST Based Web Services Using Software Evaluation Metrics. Riga Technical University, 19, 92-97P.
- [3] Rathod, D. (2017). Performance evaluation of RESTFUL web services and SOAP / WSDL web services. *International Journal of Advanced Research in Computer Science*, 8(7), 415–420. <https://doi.org/10.26483/ijarcs.v8i7.4349>
- [4] Representational State Transfer – Wikipedie [online] [cit. 28.10.2022] Dostupné z: https://cs.wikipedia.org/wiki/Representational_State_Transfer
- [5] ČÚZK - Webové služby dálkového přístupu. ČÚZK - Úvod [online]. Copyright © [cit. 31.10.2022]. Dostupné z: <https://www.cuzk.cz/Katastr-nemovitosti/Poskytovani-udaju-z-KN/Dalkovy-pristup/Webove-sluzby-dalkoveho-pristupu.aspx>
- [6] Al-Fayoumi, M., Hamad, R. H., & Al-Sarairah, J. (2018). SSOAM: Automated Security Testing Framework for SOA middleware in banking domain. *Journal of Computer Science*, 14(7), 957–968. <https://doi.org/10.3844/jcssp.2018.957.968>
- [7] Modi, Kirit & Garg, Sanjay. (2015). Discovery and Composition of Link Open Data based RESTful Web services. *International Journal of Computer Science & Communication*. 6. 294-304. 10.090592/IJCSC.2015.631.
- [8] Gamma, E., Helm, R., & Johnson, R. (1998). *Design patterns elements of Reusable Object Oriented Software*. Addison Wesley.