



Prepojenie XSD s centrálnym modelom údajov verejnej správy

Marek Šurek, marek.surek@gmail.com

26.5.2017

Úvod

- ▶ Ako smerovať štandardizáciu v oblasti schém
- ▶ Zameranie sa na centrálny dátový model s využitím sémantiky
- ▶ Cieľ je ukázať možnú cestu v dátovej interoperabilite
- ▶ Dokument ukazuje rôzne možnosti ako sa tam dopracovať, nie je to schvaľovací materiál
- ▶ Dôležitým medzníkom je schvaľovanie naplánované na 30.5, ktoré vytvára prvotný predpoklad aby mohli byť nasledovné vízie naplnené
- ▶ Využitie sémantiky a linkovaných dát v XSD schémach
 - ▶ Automatizovaná validácia
 - ▶ Export kvalitných datasetov
 - ▶ Dáta pre vyhľadávanie pre slovensko.sk



Súčasný stav

- ▶ Existujúce XSD schémy treba kopírovať do vlastných schém
- ▶ Dátová integrácia
 - ▶ na úrovni spoločných XSD schém, kde samotné formuláre neprepoužívajú namespaces spoločných schém nie je ideálna
 - ▶ je riešená textovým popisom v xs:documentation a nie je možné ju rozumne automatizovane validovať
- ▶ Treba hľadať spôsob ako riešiť centrálny model údajov (nástupca KDP) v čase keď KDP bol prekonaný -> štandard treba ďalej rozvíjať a synchronizovať s novým smerovaním



Sémantické štandardy – nastavba k XSD

- Využitie centrálného modelu údajov verejnej správy na dosiahnutie dátovej interoperability
- XSD schémy by mali priamo vychádzať resp. odkazovať centrálny model údajov vyjadrený cez URI identifikátory
- Jednotný spôsob popisovania služieb, formulárov, schém – dátová interoperabilita na všetkých úrovniach
- Využitie referenčného registra URI – MetaIS
- Existuje viacero spôsobov ako ju zabezpečiť s rôznymi úrovňami výsledku

1. Kombinácia XSD + RDF - 1

■ Existujúci stav všeobecná schéma

```
<xs:element name="FormattedName" type="FormattedNameType" minOccurs="0" maxOccurs="1" nillable="true">
  <xs:annotation>
    <xs:documentation>D.1.1.1.1 Formatovane meno vratane predpony a pripony.</xs:documentation>
  </xs:annotation>
</xs:element>
```

■ Navrhovaný stav všeobecnej schémy

■ Doplnenie mapovania cez RDF (sameAs resp. subPropertyOf)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
...
<xs:element name="FormattedName" type="FormattedNameType" minOccurs="0" maxOccurs="1" nillable="true" >
  <xs:annotation>
    <xs:appinfo>
      <owl:DataProperty ID="FormattedName">
        <rdfs:subPropertyOf rdf:resource="https://data.gov.sk/def/ontology/physical-person/formattedName"/>
        <owl:sameAs rdf:resource="https://data.gov.sk/def/ontology/physical-person/formattedName"/>
      </owl:DataProperty>
    </xs:appinfo>
  </xs:annotation>
</xs:element>
...
</xs:schema>
```

1. Kombinácia XSD + RDF - 2

- ▶ Sémantickým popisovaním je možné dosiahnuť
 - ▶ Každý si môže používať také elementy v takých početnostiach ako potrebuje
 - ▶ Nutnosť ukazovať na centrálny model
 - ▶ Dátová interoperabilita metadát je zachovaná
- ▶ Začlenenie prebieha cez vloženie RDF mapovania priamo do XSD do čast appinfo s mapovaním na konkrétny element

2. Štandardizované atribúty v rámci XSD

- Dátovú interoperabilitu je možné definovať aj na inom mieste
- Vytvorenie štandardu všeobecnej schémy na popisovanie prvkov v XSD
- Namespace : <https://data.gov.sk/def/xsd/semantic-mapping/1.0>
- Integrácia do schémy cez definovanie štandardizovaných atribútov tj.

```
<xs:complexType name="PersonNameType" classURI="https://data.gov.sk/def/ontology/physical-person/PhysicalPerson">
  <xs:sequence>
    <xs:element name="GivenName"
      type="GivenNameType"
      maxOccurs="unbounded"
      propertyURI="https://data.gov.sk/def/ontology/physical-person/givenName"/>
    <xs:element name="FamilyName"
      type="FamilyNameSCType"
      maxOccurs="unbounded"
      propertyURI="https://data.gov.sk/def/ontology/physical-person/familyName" />
  </xs:sequence>
</xs:complexType>
```

- Implementácia je priamočiara so zachovaním voľnosti pre konkrétne potreby
- Set mapovacích atribútov by bol väčší ako je na tomto jednoduchom príklade cca 8

3. Externé mapovanie elementov XSD schémy

- Mapovanie mimo XSD schému
- Referencia prvkov cez spojenie URI namespace formuláru + Xpath na daný element
- Nutnosť vytvorenia externého úložiska :
 - .owl súboru súčasť definičného balíčka formulára v .zip súbore alebo
 - mapovanie je uložené len v Metals a integráciou je šírený do iných systémov
- Možnosť rôzneho stupňa mapovania – rôzne výrazové prostriedky (rovnaké, podtrieda,...)

Príklad mapovania vlastnosti medzi XML a centrálnym modelom :

- http://schemas.gov.sk/form/30807484.ziadost_o_vydanie_potvrdenia_o_nedoplatkoch_na_socialnom_poisten_i_fo.sk/1.1/Applicant/PersonData/PhysicalPerson/PersonName/GivenName owl:sameAs pper:givenName
- http://schemas.gov.sk/form/30807484.ziadost_o_vydanie_potvrdenia_o_nedoplatkoch_na_socialnom_poisten_i_fo.sk/1.1/Applicant/PersonData/PhysicalPerson/PersonName/GivenName rdfs:subPropertyOf pper:givenName

Príklad mapovania triedy medzi XML a centrálnym modelom :

- http://schemas.gov.sk/form/30807484.ziadost_o_vydanie_potvrdenia_o_nedoplatkoch_na_socialnom_poisten_i_fo.sk/1.1/Applicant/PersonData/PhysicalPerson owl:sameAs pper:PhysicalPerson
- http://schemas.gov.sk/form/30807484.ziadost_o_vydanie_potvrdenia_o_nedoplatkoch_na_socialnom_poisten_i_fo.sk/1.1/Applicant/PersonData/PhysicalPerson owl:equivalentClass pper:PhysicalPerson
- http://schemas.gov.sk/form/30807484.ziadost_o_vydanie_potvrdenia_o_nedoplatkoch_na_socialnom_poisten_i_fo.sk/1.1/Applicant/PersonData/PhysicalPerson rdfs:subClassOf pper:PhysicalPerson

Transformácia ontológií do XSD

- Mapovanie môže prebiehať aj tvorbou 1:1 XSD k ontológií vyjadrujúcej centrálny model
- Elementy v XSD schéme sú preto priamo naviazané na centrálny model

Príklad z ontológie <https://data.gov.sk/def/ontology/physical-person> (prefix pper)

```
<owl:Class rdf:about="&pper;PhysicalPerson">
  <rdfs:label xml:lang="sk">Fyzická osoba</rdfs:label>
</owl:Class>
<owl:DatatypeProperty rdf:about="&pper;givenName">
  <rdfs:label xml:lang="sk">meno</rdfs:label>
  <rdfs:domain rdf:resource="&pper;PhysicalPerson"/>
  <rdfs:range rdf:resource="&xsd;string"/>
</owl:DatatypeProperty>
```

Transformácia ontológií do XSD

Príklad z transformácie do XSD

```
<xs:complexType name="PhysicalPerson">  
  <xs:sequence>  
    <xs:element name="GivenName" type="xs:string"/>  
  </xs:sequence>  
</xs:complexType>
```

4. Transformácia na úrovni dát

- ▶ Využitie štandardu <https://www.w3.org/TR/r2rml/> rozšíreného o generalizáciu
- ▶ Nepracujeme na úrovni XSD schémy ale na úrovni výstupného XML súboru a Xpath
- ▶ Okrem XML je podpora aj pre JSON a CSV transformácie do linked data
- ▶ Ideálne pre zabezpečenie 4 hviezdíčiek minimálne pre otvorené dáta
- ▶ Nevýhoda je nutnosť vytvorenia externého vizuálneho nástroja, ktorý bude jednoducho použiteľný
 - ▶ pre popis služieb (OpenAPI specs je zapísaná v JSONe)
 - ▶ Pre popis dát (XML, CSV, JSON) – odstránenie 3 hviezdíčkových dát

4. Transformácia na úrovni dát – XML dáta

```
<Applicant>
  <PersonData>
    <PhysicalPerson>
      <PersonName>
        <GivenName>Jozef</GivenName>
        <FamilyName>Kováč</FamilyName>
      </PersonName>
    </PhysicalPerson>
  </PersonData>
  <ID>
    <IdentifierType>
      <Codelist>
        <CodelistCode>CL004001</CodelistCode>
        <CodelistItem>
          <ItemCode>https://data.gov.sk/def/identifier-type/9</ItemCode>
          <ItemName>Rodné číslo</ItemName>
        </CodelistItem>
      </Codelist>
    </IdentifierType>
    <IdentifierValue>8710096542</IdentifierValue>
  </ID>
</Applicant>
```

4. Transformácia na úrovni dát – Príklad definície mapovania

```
<#PersonMapping>
  rml:logicalSource [
    rml:source "data.xml" ;
    rml:referenceFormulation ql:XPath
  ];
  rr:subjectMap [
    rr:class pper:PhysicalPerson
    rr:template "https://data.gov.sk/id/physical-person/{/Applicant/ID/IdentifierValue}"
  ];
  rr:predicateObjectMap [
    rr:predicate pper:givenName;
    rr:objectMap [
      rml:reference "/Applicant/PersonData/PhysicalPerson/PersonName/GivenName";
      rr:datatype xsd:string
    ]
  ];
].
```

4. Transformácia na úrovni dát – výstup transformácie

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:ppper="https://data.gov.sk/def/ontology/physical-person">

  <ppper:PhysicalPerson rdfs:about="https://data.gov.sk/id/physical-person/8710096542">
    <ppper:givenName xml:lang="sk">Jozef</ppper:givenName>
  </ppper:PhysicalPerson>

</rdf:RDF>
```

Zhodnotenie možností

- ▶ Zabezpečenie typovej konzistencie cez 1:1 ontológie vs xsd schémy
- ▶ Štandard XSD by mal byť čo najviac v synchronizácii s K9.4 Lepšie dáta
- ▶ Navrhované možnosti riešia
 - ▶ Dátovú interoperabilitu
 - ▶ Tranformáciu dát
 - ▶ Interoperabilná metadáta o formulároch – vyhľadávanie na slovensko.sk s oveľa lepšími dátami

Krátke zhodnotenie možností

➤ Možnosť 1

➤ Pre

- najbližšie k developerom – práca len so známou XSD
- Pridávajú mapovanie na centrálny model

➤ Proti

- Nie je možná automatizovaná priamočiara transformácia dát do sémantiky – centrálny model je len popisný
- Nutná znalosť jazyka RDF na dopĺňanie

➤ Možnosť 2

➤ Pre

- najbližšie k developerom – práca len so známou XSD
- Pridávajú mapovanie na centrálny model
- Na mapovanie sa využívajú xml atribúty (nie je nutné poznať RDF)

➤ Proti

- Nie je možná automatizovaná priamočiara transformácia dát do sémantiky – centrálny model je len popisný

Krátke zhodnotenie možností 2

Možnosť 3

Pre

- XSD schéma zostáva nezmenená
- Mapovanie prebieha cez externý systém (klikátko) alebo napísaním RDF mapovania ručne do súboru s následnou validáciou

Proti

- Mapovanie prebieha cez externý systém (klikátko) alebo napísaním RDF mapovania ručne do súboru s následnou validáciou (treba vyvinúť takýto systém a implementovať ho do MEF eForm designera)
- Nie je možná automatizovaná transformácia dát do sémantiky (iba popisné)
- Dáta nie sú len v XSD ale mapovanie je v externom zdroji

Možnosť 4

Pre

- Ideálne riešenie pre automatizovanú transformáciu do sémantiky (najkomplexnejšie riešenie, ktoré vie vyriešiť ako tagovanie tak aj transformáciu)
- XSD je bez zmeny
- Nadstavba k možnostiam 1-3 (je možné používať ako kombináciu resp. dobrovoľné rozšírenie možností 1-3)
- Štandardizované previazanie XSD, JSON (definícia služieb), CSV – dáta v službách ako aj v otvorených dátach vedia byť vždy transformované do centrálného modelu údajov automatizovane

Proti

- Dodatočná „byrokracia“, ktorú je nutné robiť v externom systéme
- Nutné vytvoriť externý systém (klikátko) pre čo najjednoduchšiu interakciu



Záver

- Sú navrhované riešenia rozširovania sémantickej reprezentácie do praxe
- Opierajú sa o oporné body architektúry tj.
 - Centrálny model údajov verejnej správy pomocou URI a ontológií
 - Metals
- Schvaľovanie v pondelok 30.5 vytvára len podhubie

Backup slide

Codelist element pre firmu s využitím URI

- Nahradenie textových identifikátorov pre položky jednoznačným URI identifikátorom zaregistrovaným v Metals

```
<ID>
  <IdentifierType>
    <Codelist>
      <CodelistCode>https://data.gov.sk/def/codelist/CL004001</CodelistCode>
      <CodelistItem>
        <ItemCode> https://data.gov.sk/def/identifier-type/7</ItemCode>
        <ItemName>Identifikačné číslo</ItemName>
      </CodelistItem>
    </Codelist>
  </IdentifierType>
  <IdentifierValue>https://data.gov.sk/id/legal-subject/12345678</IdentifierValue>
</ID>
```