

Tvorba DSD a kódovníků pro fiskální data z projektu OpenBudgets.eu

Prof. Ing. Vojtěch Svátek, Dr.

Zimní semestr 2019

<http://nb.vse.cz/~svatek/rzzw.html>

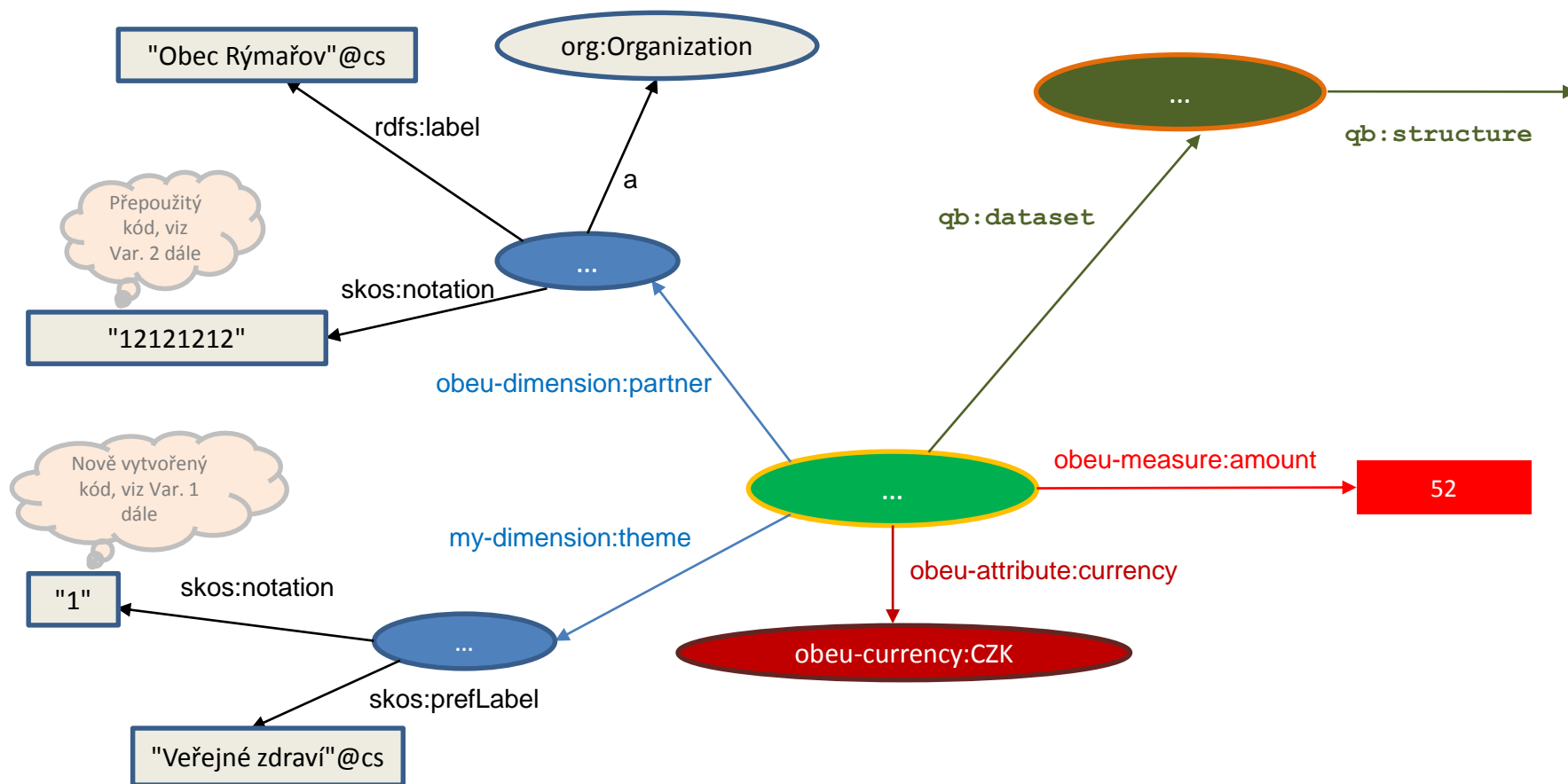
Základní info

- Předpoklad: vstupní data jsou v CSV npod., a existuje k nim aspoň minimální textová dokumentace
- Postup je popsán v projektovém dokumentu *D1.5 - Final release of data definitions for public finance data*
 - Komponenty, kódovníky, příklady použití, včetně návodu a širšího úvodu (i k DCV)
<http://openbudgets.eu/assets/deliverables/D1.5.pdf>
- Řada doporučení je uvedena v pracovním materiálu MV ČR
 - <https://data.gov.cz/otevřené-formální-normy/propojená-data/draft/>
- Zbývající část prezentace shrnuje
 - ukázkový příklad a možné vzory pro vytváření dat v RDF
 - časté chyby, mj. podle zpětné vazby z výuky 4IZ440 2016-18

Ukázkový příklad – vstupní CSV

ID	Název projektu	IČO Příjemce	Název příjemce	Tematická oblast	Úhrn dotace (tis. Kč)
1	Nákup kapesníků	12121212	Obec Rýmařov	Veřejné zdraví	52
2	Nákup kapesníků	13131313	Obec Kašlany	Veřejné zdraví	1021
3	Nákup kapesního roboty na skládání kapesníků	45454545	KapTex, s.r.o.	Rozvoj konkurenceschopnosti	65
4	Solární stanice pro roboty	45454545	KapTex, s.r.o.	Udržitelné zdroje energie	120

Ukázkový příklad – grafová struktura pozorování



„Kuchařka“ modelování kódovníků

Obecné principy (platné pro SKOS)

- Každý kód dimenze/atributu v datech by měl být explicitně identifikovaný jako instance *skos:Concept*, a měl by mít textovou reprezentaci, nejlépe jako *skos:prefLabel*
- Vhodné je převzít/zavést explicitní kódovník, ke kterému je kód přiřazen pomocí *skos:inScheme*, a k IRI kódu doplnit jeho zápis v podobě literálu, jako hodnotu *skos:notation*
- Z ne-RDF číselníků přebíráme kódy, z nichž vyrobíme IRI slepením se jmenným prostorem vzniklého RDF kódovníku

„Kuchařka“ modelování kódovníků

Varianta 1: kódovník neexistuje v žádné podobě

- Vyjdeme jen z textové hodnoty, která je ve vstupních datech, a pro každé pozorování přiřadíme hodnotu komponenty (dimenze resp. atributu) jako:

<pozorování> <komponenta> [a skos:Concept ;
skos:prefLabel **<textová hodnota>**] .

- Tj. zapíšeme kód buď jako bnode, nebo např. jako „nečitelné“ IRI vytvořené pomocí SHA1 transformace z hodnoty prefLabel

„Kuchařka“ modelování kódovníků

Varianta 1 - příklad

- V datech je sloupec „Účel dotace“, který je při převodu do RDF interpretován jako *obeu-dimension:programmeClassification* (existující dimenze z OBEU modelu
 - ještě lepší by bylo vytvořit její podvlastnost s vlastním kódovníkem, tak i dále
- Hodnota sloupce pro daný řádek je „1“
- Část popisu pozorování v RDF pak může pro tento řádek vypadat takto:

```
<obs1234> obeu-dimension:programmeClassification  
[ a skos:Concept ; skos:notation "1" ] .
```

„Kuchařka“ modelování kódovníků

Varianta 2: kódovník existuje, ale není v RDF

- Přepoužijeme známý zápis kódu v pozorování, zpravidla ve struktuře:
<pozorování> <komponenta> <IRI kódovníku>/<zápis kódu> .
- IRI kódovníku zpravidla navrhne podle jeho názvu v původním zdroji
- Ke každému IRI kódu (složenému z IRI kódovníku a ze zápisu kódu) explicitně přiřadíme jeho zápis

„Kuchařka“ modelování kódovníků

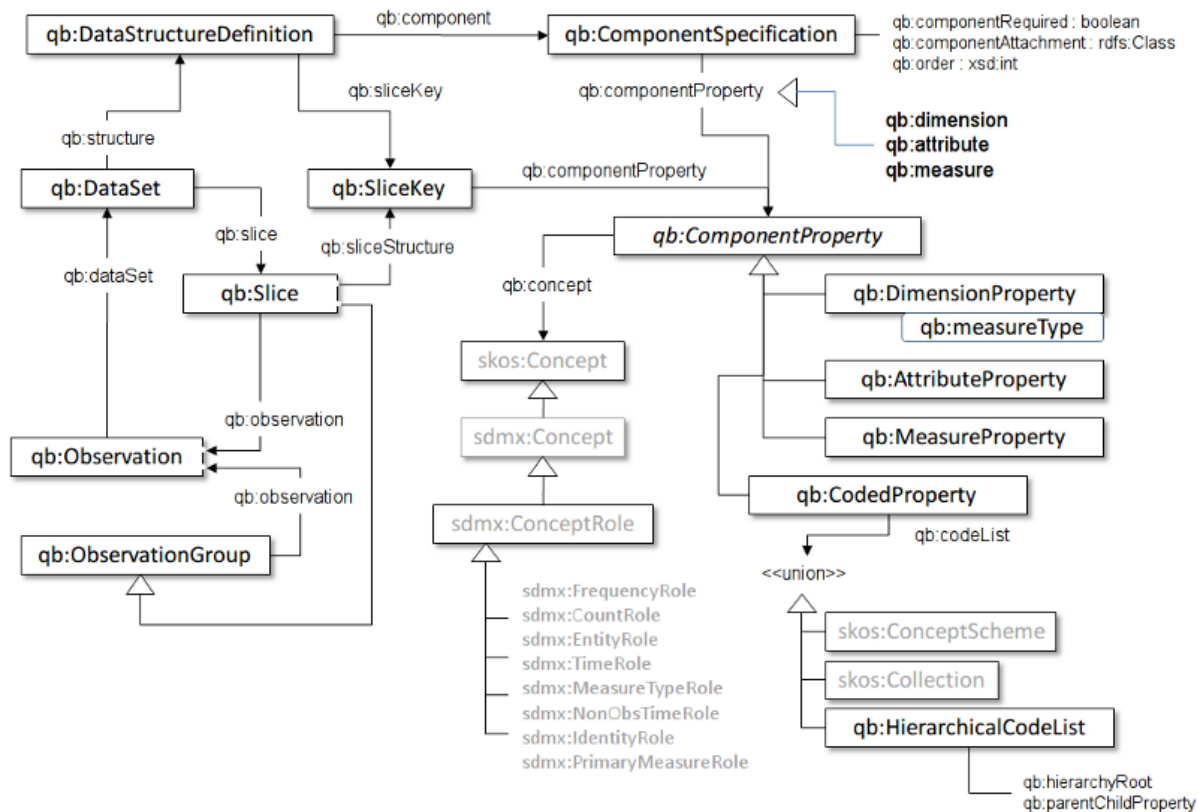
Varianta 2 - příklad

- Sloupec „Účel dotace“ má přiřazený tabulkový kódovník označený „Programová priorita“, zahrnující i kód „1“ s textovým popisem „Stavební práce“
- Kódovníku navrhne IRI z vhodného jmenného prostoru; pro data projektu OBEU (ovšem ne studentskou práci) např.
`<http://data.openbudgets.eu/resource/codelist/program-priority>`
- Jednotlivý kód bude definován jako
`<http://data.openbudgets.eu/resource/codelist/program-priority/1>`
a `skos:Concept` ; `skos:prefLabel` „Stavební práce"@cs ; `skos:notation` "1" ;
`skos:inScheme` `<http://data.openbudgets.eu/resource/codelist/program-priority>`
- Část popisu pozorování v RDF pak zapíšeme jako
`<obs1234> obeu-dimension:programClassification`
`<http://data.openbudgets.eu/resource/codelist/program-priority/1>`

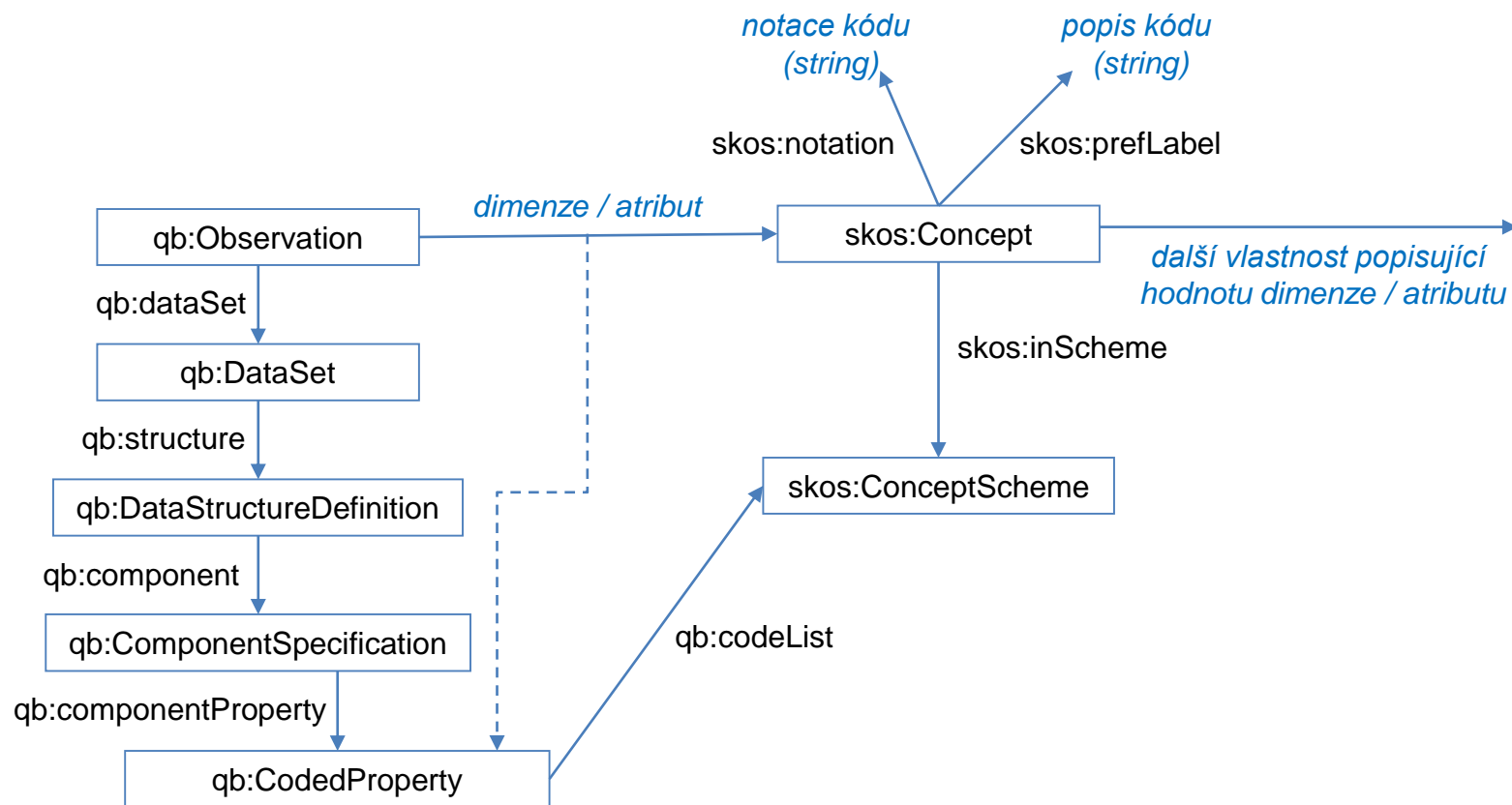
„Kuchařka“ modelování kódovníků

- **Varianta 3: kódovník již existuje v RDF**
 - IRI kódovníku i zápis kódu (dohromady tvořící IRI kódu) převezmeme z původního zdroje
 - Pokud nemůžeme spoléhat na to, že si aplikace budou (moci) číselník vždy dereferencovat, doplníme do dat vlastnosti kódů jako ve Variantě 2

Schéma DCV (pro osvěžení)



Možné schéma využití DCV+SKOS



Kontrolní seznam pro DSD, komponenty a dataset (1)

- Není možné *přepoužít* existující komponenty (obvykle SDMX nebo OBEU) spíš než navrhovat nové?
- Případně, není možné nové komponenty *podřadit* existujícím jako speciální případy?
- Jsou nové komponenty navrženy ve vhodném *jmenném prostoru*? Nejde o „namespace hijacking“?
 - Nejvhodnější využít pro data i DSD a kódovníky jmenné prostory v rámci domény vyčleněné katedrou
http://lod.kizi.vse.cz/4iz440/zs2018/Tym<číslo_týmu>/něco

Kontrolní seznam pro DSD, komponenty a dataset (2)

- Nepředstavují některé navrhované dimenze jen různé charakteristiky *stejného* objektu (např. organizace, tematické oblasti apod.), o kterém vypovídá pozorování?
 - Pokud ano, spíše připojit další vlastnosti objektům – hodnotám dimenzí – mimo dimenzionální schéma (tj. vytvořit „model sněhové vločky“ spíše než „hvězdy“)
- Jsou dimenze navrženy jako *objektové* vlastnosti, z hlediska range? Jaký má každá vlastnost *kódovník*? A jsou odpovídajícím způsobem zapsána i *data* v datasetu?
 - Pozor na nutnost transformace řetězců na IRI v rámci ETL

Kontrolní seznam pro DSD, komponenty a dataset (3)

- Nabývají míry *kvantitativních* hodnot?
 - Kvalitativní hodnoty jsou spíš kandidáty na modelování pomocí dimenze (i když existují výjimky)
- Jsou všechny části DSD a dat korektně *propojeny*?
 - Pozorování s datasetem
 - Dataset s DSD
 - DSD s komponentami
 - Komponenty s kódovnicí
- Jsou všechny dimenze a míry přítomny u *všech* pozorování? Má každé pozorování unikátní IRI, a jen *jednu* hodnotu pro každou komponentní vlastnost?