ČNB ČESKÁ NÁRODNÍ BANKA

Věcné zadání projektu SDAT

B - Metapopis

Červenec 2016

Obsah

1 Úvod ... 8
2 Obecný úvod do metapopisu ... 8
2.1 Údaj jako základní prvek metapopisu ... 8
2.2 Sledování historie instancí objektů metapopisu ... 10
2.2.1 Číslo verze a varianty instance objektu ... 11
2.2.2 Objektový model pro sledování historie instancí objektů ... 11
2.2.3 Přístup „Bez sledování historie“ ... 11
2.2.4 Přístup „Bez sledování historie – stavy“ ... 11
2.2.5 Přístup „Sledování historie – časová platnost“ ... 11
2.2.6 Přístup „Sledování historie – časová platnost + stavy“ ... 11
2.2.7 Přístup „Sledování historie – časová platnost na každém atributu“ ... 11
2.3 Vazby mezi jednotlivými objekty ... 11
2.3.1 Objekt #Objekt ... 12
2.3.2 Objekt #ObjektAtribut ... 13
2.3.3 Objekt #ObjektZávislost ... 13
2.3.4 Dopad editace instance objektu na číslo verze a varianty ... 16
2.3.5 Objektový model ... 18
2.3.6 Rámcové vymezení závislosti objektů ... 18
2.4 Objekty metapopisu ... 20
2.4.1 Základní vlastnosti objektů ... 20
2.4.2 Standardní atributy objektů ... 21
2.4.3 Základní principy práce s objekty ... 23
2.4.4 Životní cyklus objektu ... 24
2.4.5 Základní pravidla práce s instancemi objektů ... 27
2.5 Kontrola konzistence ... 29
3 Objektový model ... 31
3.1 Objekt Vykazovací rámec ... 31
3.1.1 Objekt Metodika vykazovacího rámce ... 31
3.1.2 Objekt Výkaz ve Vykazovacím rámci ... 33
3.1.3 Objekt Verze výkazu v Metodice vykazovacího rámce ... 34
3.1.4 Objekt Štítky výkazu ... 34

3.2 Objekt Výkaz 35
3.2.1 Atributy objektu Výkaz 36
3.3 Objekt Blok výkazu 36
3.3.1 Atributy objektu Blok výkazu 37
3.4 Objekt Datová oblast 37
3.4.1 Atributy objektu Datová oblast 39
3.5 Objekt Číselník 40
3.5.1 Atributy objektu Číselník 42
3.6 Objekt Položka číselníku 43
3.6.1 Atributy objektu Položka číselníku 44
3.6.2 Dynamické atributy objektu Položka číselníku 44
3.7 Objekt Hierarchie číselníku 46
3.7.1 Atributy objektu Hierarchie číselníku 48
3.7.2 Proces tvorby Hierarchií číselníku 48
3.8 Objekt Položka hierarchie 52
3.8.1 Atributy objektu Položka hierarchie 53
3.9 Objekt Doména číselníku 53
3.9.1 Atributy objektu Doména číselníku 56
3.10 Objekt Převodník 57
3.10.1 Atributy objektu Převodník 57
3.11 Objekt Položka převodníku 58
3.12 Objekt Účtová osnova 58
3.12.1 Atributy objektu Účtová osnova 59
3.13 Objekt Účet 59
3.13.1 Atributy objektu Účet 61
3.14 Objekt Datový typ 61
3.14.1 Atributy objektu Datový typ 62
3.15 Objekt Doména datového typu 62
3.15.1 Atributy objektu Doména datového typu 63
3.16 Objekt Ukazatel 63
3.16.1 Objekt Dodatečná konkretizace Ukazatele 64
3.16.2 Objekt Ukazatel v Datové oblasti 67
3.16.3 Atributy objektu Ukazatel 69

3.17 Objekt Parametr ... 69
3.17.1 Atributy objektu Parametr ... 71
3.18 Objekt Údaj ... 71
3.18.1 Vznik Údaje – statická Datová oblast ... 72
3.18.2 Popis Údaje nedimenzionálním parametrem ... 76
3.18.3 Atributy objektu Údaj ... 78
3.18.4 Dynamické atributy Údaje ... 79
3.19 Objekt Kontrola ... 80
3.19.1 Atributy objektu Kontrola ... 81
3.19.2 Způsob vytvoření kontroly ... 82
3.19.3 Objekt Jednovýkazová kontrola (JVK) ... 82
3.19.4 Objekt Kontrola časové řady (KČŘ) ... 84
3.19.5 Objekt Mezivýkazová kontrola (MVK) ... 85
3.19.6 Odchylka v sémantických kontrolách ... 86
3.20 Objektový model – Definice kontrol ... 88
3.21 Objektový model – Metapopis ... 89
4 Knihovna ... 90
5 Hlavní procesy ... 90
5.1 Proces tvorby metapopisu ... 90
5.1.1 Spouštěč procesu ... 91
5.1.2 Průběh procesu ... 91
5.1.3 Výstup procesu ... 92
5.2 Proces definice Metodiky vykazovacího rámce ... 92
5.2.1 Spouštěč procesu ... 93
5.2.2 Průběh procesu ... 93
5.2.3 Výstup procesu ... 94
5.3 Proces tvorby Výkazu ... 94
5.3.1 Spouštěč procesu ... 96
5.3.2 Průběh procesu ... 96
5.3.3 Výstup procesu ... 101
5.4 Proces tvorby objektů popisujících údaje ... 101
5.4.1 Spouštěč procesu ... 102
5.4.2 Průběh procesu ... 102

5.4.3 Výstup procesu 106
5.5 Proces tvorba kontrol výkazu 107
5.5.1 Spouštěč procesu 108
5.5.2 Průběh procesu 108
5.5.3 Výstup procesu 112
5.6 Proces schválení výkazů v Metodice vykazovacího rámce 112
5.6.1 Spouštěč procesu 113
5.6.2 Průběh procesu 113
5.6.3 Výstup procesu 114
5.7 Proces prezentace Výkazu 114
5.7.1 Spouštěč procesu 115
5.7.2 Průběh procesu 115
5.7.3 Výstup procesu 118
6 Podpůrné procesy 118
6.1 Proces Přebíraní metapopisu z externích zdrojů 118
6.1.1 Přebírání metapopisu z XBRL taxonomie 118
6.1.2 Spouštěč procesu 120
6.1.3 Průběh procesu 120
6.1.4 Výstup procesu 120
6.2 Proces Navazování časových řad Údajů 121
6.2.1 Spouštěč procesu 121
6.2.2 Průběh procesu 121
6.2.3 Výstup procesu 122
7 Funkční požadavky 123
7.1 Obecné požadavky pro objekty metapopisu 123
7.2 Vykazovací rámec 128
7.3 Metodika vykazovacího rámce 128
7.4 Výkaz ve Vykazovacím rámci 132
7.5 Verze výkazu v Metodice vykazovacího rámce 133
7.6 Štítky výkazu 133
7.7 Výkaz 134
7.8 Blok výkazu 141
7.9 Datová oblast 146

7.10 Číselník, Položka číselníku ... 155
7.11 Hierarchie číselníku, Položka hierarchie ... 162
7.12 Doména číselníku ... 171
7.13 Převodník, Položka převodníku ... 177
7.14 Účtová osnova, účet ... 180
7.15 Datový typ ... 184
7.16 Doména datového typu ... 187
7.17 Ukazatel, Konkretizace ukazatele, Dodatečná konkretizace ukazatele ... 189
7.18 Parametr, Konkretizovaný parametr ... 198
7.19 Údaj ... 204
7.20 Jednovýkazová kontrola (JVK) ... 207
7.21 Mezivýkazová kontrola (MVK) ... 215
7.22 Kontrola časové řady (KČŘ) ... 223
7.23 Knihovna ... 227
7.24 Navazování časových řad ... 228
7.25 Grafické zobrazení struktury ... 230
7.26 Načítání artefaktů z externího metapopisu ... 231
7.27 Speciální kontroly (MKT) ... 232
8 Přílohy ... 233
8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systém SDAT 233
8.1.1 Kontroly v časové řadě (KČŘ) ... 233
8.1.2 Kontroly v systému kapitálových trhů (MKT) ... 234
8.2 Příloha 2 - Příklady pro pravidla aktualizace a synchronizace Hierarchií číselníku 244
8.2.1 Pravidlo 1 — Vložení elementární Položky hierarchie do Hierarchie číselníku 244
8.2.2 Pravidlo 2 — Smazání elementární Položky hierarchie z Hierarchie číselníku 245
8.2.3 Pravidlo 3 — Smazání uzlové položky z Hierarchie číselníku ... 246
8.2.4 Pravidlo 4 — Smazání uzlové položky z Hierarchie číselníku včetně jejího elementárního obsahu ... 247
8.2.5 Pravidlo 5 — Vložení uzlové položky do Hierarchie číselníku ... 248
8.2.6 Pravidlo 6a — Změna uzlové položky na elementární položku ... 250
8.2.7 Pravidlo 6b — Změna elementární položky na uzlovou položku ... 251

8.2.8 Pravidlo 7 — Smazání celého uzle hierarchie včetně uzlové položky 252
8.2.9 Konflikt typu Duplicita v nezávislých uzlech 253
8.2.10 Konflikt typu Duplicita v závislých uzlech 254
8.3 Příloha 3 - Příklady pro pravidla aktualizace Domén číselníku 254
8.3.1 Pravidlo 1 254
8.3.2 Pravidlo 2 255
8.3.3 Pravidlo 3 255
8.3.4 Pravidlo 4 256
8.4 Příloha 4 — Počty instancí vybraných objektů metapopisu v roce 2016 v systému MtS 257

1 Úvod

Tento dokument se zabývá základním prvkem systému, kterým je tzv. metapopis informací, podle kterého vykazující osoby předkládají České národní bance data hlášení a výkazů. Cílem metapopisu je jednoznačné vymezení metapopisu jednotlivých údajů tak, aby vykazující osoby byly schopny přiřadit odpovídající hodnoty ze svých interních informačních systémů a aby byla zajištěna srovnatelnost mezi jednotlivými vykazujícími osobami.

Jednotlivé části obsahují popis typů informací, které jsou do ČNB zasílány, definice jednotlivých objektů používaných pro metapopis vč. jejich společných i specifických atributů, životního cyklu apod. ve formě objektového modelu (Class diagram) a popis procesů, v rámci kterých vzniká metapopis, a které by měly být automatizovány.

2 Obecný úvod do metapopisu

Metapopis představuje jádro systému SDAT. Metapopis je sada objektů a nad nimi vybudované funkcionality, jejichž primárním cílem je dosáhnout podrobného a strukturovaného popisu Údaje. Údaj, a především jeho zcela přesný a strukturovaný popis, pak tvoří základní prvek sběru dat od vykazujících osob.

Strukturovaný popis Údaje je založen na jasně definovaných pravidlech. Popis Údajů se provádí prostřednictvím tzv. metadat, jejichž účelem je popsat jednotlivý zjišťovaný Údaj vyčerpávajícím způsobem tak, aby bylo tento popis možno použít pro automatizaci jak na straně vykazujících osob, tak na straně ČNB, např. pro potřeby dalšího zpracování došlých dat pro mezinárodní organizace aj.

2.1 Údaj jako základní prvek metapopisu

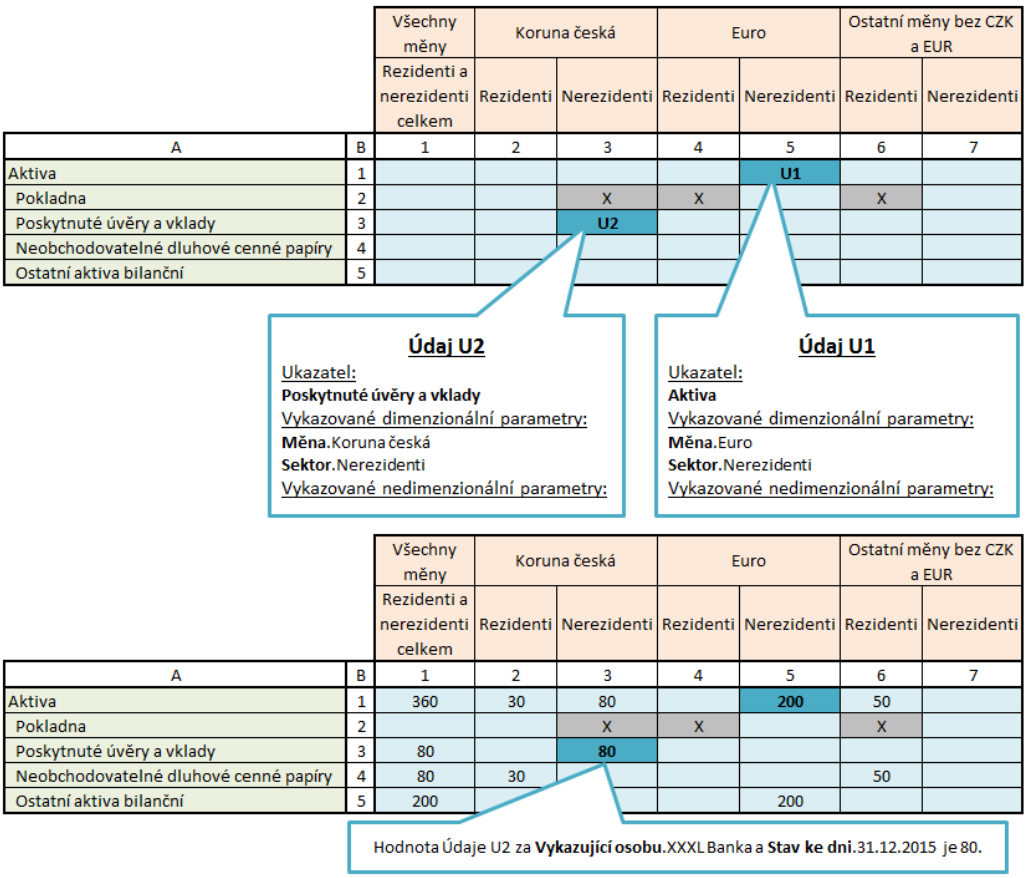
Elementárním prvkem metapopisu je objekt Údaj, který představuje jednotlivé pole výkazu. Údaj musí být popsán tak, aby bylo zaručeno, že jednotlivé vykazující osoby vykáží takovou Hodnotu údaje, jaká je očekávána. Údaj je popsán pomocí objektů metapopisu, tj. Ukazatelem a sadou jednoho nebo více Parametrů. Ukazatel vyjadřuje základní obsah sledované veličiny a Parametr(y) tento Údaj dále specifikují (tzv. konkretizují). Údaj popsaný pomocí Ukazatele a sady Parametrů tvoří dohromady popis Údaje.

Veškerá metadata týkající se údajů a dalších objektů jsou zveřejněna Vykazujícím osobám. Jakmile Vykazující osoby znají tento popis Údaje, vědí, jaká data v rámci své Vykazovací povinnosti mají poskytovat. Vykazovaná data, zasílaná do ČNB za konkrétní Vykazující osobu a stav ke dni, se nazývají Hodnota údaje (podrobně je objekt Hodnota údaje vysvětlen v dokumentu D – Sběr dat, kapitola 2.9 Objekt Hodnota údaje).

Z pohledu způsobu definice údaje rozlišujeme tři typy Údajů, tj. statický,dynamický a údaj typu soubor, které se od sebe liší rozsahem oboru hodnot parametrů:

a) statický Údaj

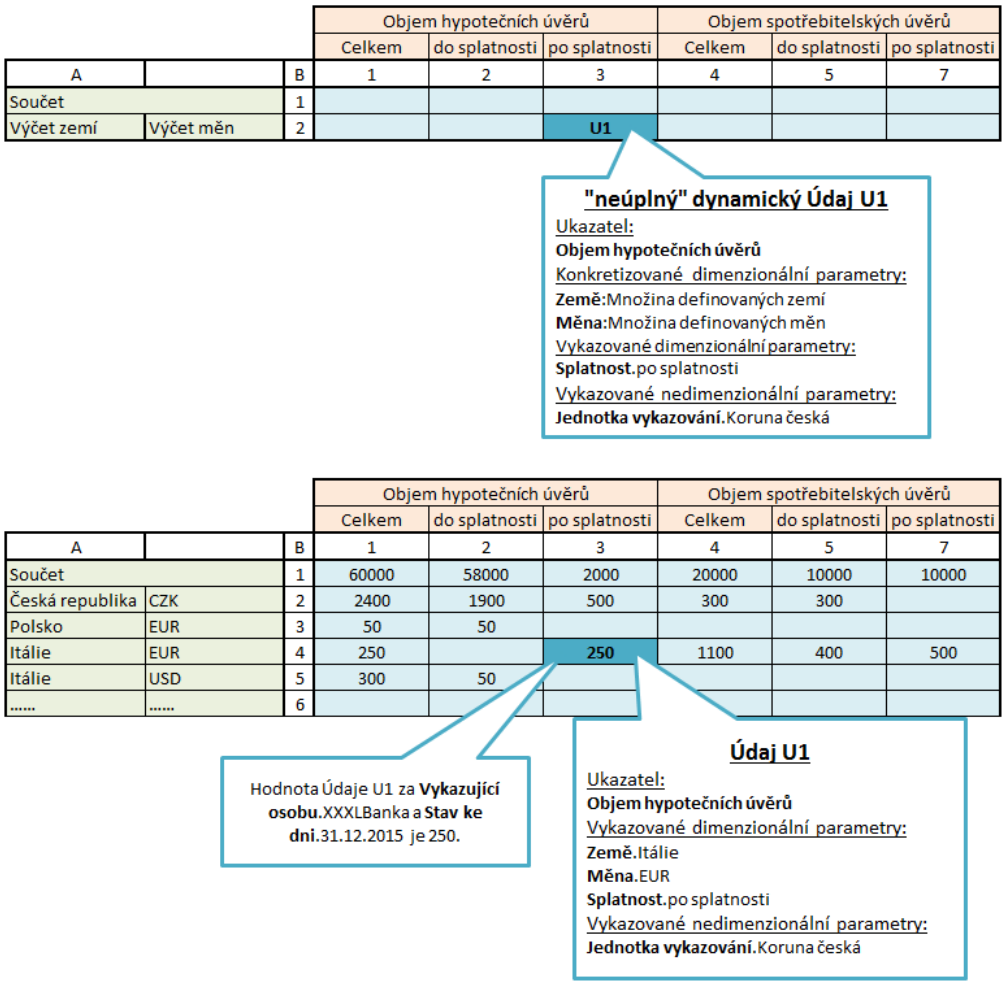
   Statický Údaj má pro každou kombinaci Ukazatel/Parametr dánu právě jednu hodnotu, kterou může nabývat. Ke statickému Údaji tak Vykazující osoba zasílá k určitému stavu ke dni právě jednu Hodnotu údaje.



Obrázek 1 - Znázornění statického Údaje a Hodnoty údaje

b) dynamický Údaj

U dynamického Údaje je hodnota Parametru (nebo více Parametrů) dána definovaným výčtem oboru hodnot nad Datovým typem, Doménou číselníku a Hierarchií číselníku, kterých může nabývat. Jedná se v podstatě o „neúplný“ Údaj. K tomuto „neúplnému“ dynamickému Údaji Vykazující osoba zasílá k určitému stavu ke dni více Hodnot údajů tak, aby byly vykázány všechny konkrétní kombinace přípustných hodnot Parametrů, které se u Vykazující osoby vyskytují.



Obrázek 2 - Znázornění dynamického Údaje a Hodnoty údaje

c) Údaj typu soubor

Údaj typu soubor je standardně popsán metadaty a jeho oborem hodnot jsou binární data.

2.2 Sledování historie instancí objektů metapopisu

Objekty metapopisu (stejně tak jako další objekty systému) podléhají základnímu pravidlu, které definuje, že instance těchto objektů se nemažou, ale ukončuje se jejich časová platnost (toto pravidlo neplatí striktně vždy, mazání je v některých případech povoleno, viz kapitola 7 Funkční požadavky). Proces ukončování časové platnosti jednotlivých instancí nazýváme také „verzováním“, protože (v některých případech) je každá instance objektu, která je časově vymezena, identifikována číslem verze a varianty.

Protože toto pravidlo je společné i pro další objekty systému, je na tomto místě uveden je stručný výčet různých přístupů k procesu sledování historie instancí objektů tak, aby se na tyto přístupy dalo odkazovat v textu od jednotlivých objektů. Detailní popis je pak uveden v dokumentu A – Obecné požadavky, kapitola 2.2 Sledování historie instancí objektů.

2.2.1 Číslo verze a varianty instance objektu

Viz dokument A – Obecné požadavky, kapitola 2.2.1 Číslo verze a varianty instance objektů.

2.2.2 Objektový model pro sledování historie instancí objektů

Viz dokument A – Obecné požadavky, kapitola 2.2.2 Objektový model pro sledování historie instancí objektů.

2.2.3 Přístup „Bez sledování historie“

Viz dokument A – Obecné požadavky, kapitola 2.2.3 Přístup „Bez sledování historie“.

2.2.4 Přístup „Bez sledování historie – stavy“

Viz dokument A – Obecné požadavky, kapitola 2.2.4 Přístup „Bez sledování historie - stavy“.

2.2.5 Přístup „Sledování historie – časová platnost“

Viz dokument A – Obecné požadavky, kapitola 2.2.5 Přístup „Sledování historie – časová platnost“.

2.2.6 Přístup „Sledování historie – časová platnost + stavy“

Viz dokument A – Obecné požadavky, kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“.

2.2.7 Přístup „Sledování historie – časová platnost na každém atributu“

Viz dokument A – Obecné požadavky, kapitola 2.2.7 Přístup „Sledování historie – časová platnost na každém atributu“ (tzv. dynamické atributy).

2.3 Vazby mezi jednotlivými objekty

Díky vysoké míře provázanosti objektů v oblasti metapopisu je třeba zajistit vzájemnou konzistenci dat. Změna platnosti instance jednoho objektu, vytvoření nové verze nebo varianty musí vyvolat odpovídající akce u souvisejících instancí objektů. Pokud by se tak nestalo, došlo by k tomu, že metadata, tolik potřebná pro správné zajištění vykazovacích povinností, nebudou konzistentní.

Aby tato konzistence byla zajištěna, musí být dodrženy následující zásady:

- závislosti mezi objekty není možné definovat staticky (na úrovni programového kódu), ale naopak je nutné, aby závislosti byly definovatelné (a v čase měnitelné) uživatelským způsobem. U objektů, které podléhají sledování historie a stavů musí být možnost

definovat závislosti s ohledem na stav, v jakém se měněná instance nachází v době provádění změny,

- pokud u nějaké instance objektu dojde k zásadní změně (změní se číslo verze nebo varianty), je nutné změnit číslo verze a varianty instance objektu i u instancí objektů, které jsou na měněné instanci objektu závislé. Systém musí umožnit definovat závislosti mezi objekty s tím, že musí být možno říci, zda a jak se změna instance jednoho objektu promítne do instancí dalších objektů (vytvoření nové verze nebo varianty závislého objektu),
- hodnota uvedená v atributu číslo verze a varianty instance objektu vypovídá o tom, jaký charakter změny je v rámci dané změny proveden:
    - pokud je **změna zásadního charakteru (tzv. „podstatná změna“)**, pak je změněno číslo verze (číslo verze a varianty instance objektu se například změní z 001.000 na 002.000),
    - pokud je **změna drobného charakteru (tzv. „méně podstatná změna“)**, pak je změněno pouze číslo varianty (číslo verze a varianty instance objektu se například změní z 001.000 na 001.001),
- ne každá změna vyvolává změnu čísla verze a varianty instance objektu; musí být možnost, aby bylo možno změnit hodnotu určitého atributu, aniž by vznikla nová instance objektu (tzv. přepis hodnoty atributu; **tzv. „nepodstatná změna“**).

O tom, zda editací instance nějakého objektu dojde k vytvoření nové verze či nové varianty (anebo změny proběhnou v rámci existující instance objektu a žádná nová verze nebo varianta nevznikne) rozhoduje uživatel. Ten před zahájením editace rozhodne, co chce udělat. Systém pouze kontroluje rozsah provedených změn a v případě, že dojde k nesouladu mezi změnou provedenou uživatelem a nastavením systému, upozorní systém na tuto skutečnost uživatele a navrhne mu jiný postup (například místo nové varianty vytvořit novou verzi, protože došlo ke změně atributu, který s ohledem na stav, ve kterém je měněná instance, vyžaduje, aby s jeho změnou došlo k vytvoření nové verze, nikoli varianty).

Za účelem splnění výše uvedených požadavků jsou do systému SDAT navrženy meta-objekty, pomocí kterých lze definovat závislosti mezi reálnými objekty/atributy metapopisu.

2.3.1 Objekt #Objekt

Účelem objektu #Objekt je zachytit seznam všech objektů systému SDAT (reálné využití je zamýšleno pouze pro objekty metapopisu) tak, aby ke každému jednomu objektu bylo možno připojit jeho atributy (definice atributů je nutná s ohledem na možnost definovat, zda změna daného atributu vyvolává změnu verze nebo změnu varianty).

Stejně tak je účelem tohoto objektu podchytit všechny objekty, pro které má smysl definovat vzájemné závislosti mezi objekty tak, aby v případě, že dojde k vytvoření nové verze nebo varianty nějakého objektu, systém věděl, pro které další instance kterých dalších objektů musí vzniknout nové verze nebo varianty.

Instancemi objektu #Objekt budou reálné objekty systému, tedy například Výkaz, Blok výkazu, Datová oblast a další.

2.3.2 Objekt #ObjektAtribut

Účelem objektu #ObjektAtribut je zachytit seznam všech atributů, které patří k související instanci objektu #Objekt. Objekt #ObjektAtribut je spojen s objektem #Objekt kompoziční vazbou s kardinalitou 1:N, tj. jedna instance objektu #Objekt může mít N instancí objektu #ObjektAtribut, ale jedna instance objektu #ObjektAtribut musí mít právě jednu související instanci třídy #Objekt. To znamená, že v rámci jednoho objektu je možno evidovat N atributů, ale konkrétní atribut smí být připojen k právě jednomu objektu. Se zánikem instance objektu #Objekt zanikají všechny související instance objektu #ObjektAtribut.

V rámci objektu #ObjektAtribut existuje atribut s názvem „změna vyvolává“, pomocí kterého je určeno, jak reálný atribut existujícího objektu SDAT reaguje na změny. Reakce atributu na změny musí být definována v kontextu konkrétního stavu dané instance (to je zachyceno pomocí atributu s názvem „stav“) v případě, že se jedná o objekt, který sleduje stavy. V případě objektu, který stavy nesleduje, bude hodnota atributu „stav“ NULL.

To znamená, že lze nastavit různé reakce jednoho a téhož atributu v závislosti na tom, v jakém stavu se nachází daná instance. **Pro jeden atribut smí být jeho chování (hodnota atributu „změna vyvolává“) pro jeden stav instance (atribut „stav“) definováno maximálně jednou**. Pokud pro nějaký atribut není chování pro nějaký stav nastaveno, pak se má za to, že změna hodnoty atributu není dovolena.

Existují tyto možnosti, které mohou být uživatelem použity pro definici hodnoty atributu „změna vyvolává“:

- změna hodnoty atributu není povolena[1],
- **změna bez změny verze nebo varianty:** změna atributu je možná bez nutnosti vytvořit nové verze a varianty (jedná se tak o tzv. „**nepodstatnou změnu**“). Toto umožní přepsat hodnotu atributu kdykoli, bez ohledu na to, zda je aktuálně platný z hlediska času nebo stavu. Předpokládá se využití pro atributy typu „popis“ nebo „poznámka“,
- **změna čísla varianty:** změna atributu vyžaduje novou variantu instance objektu. V atributu číslo verze a varianty dochází tedy ke změně hodnoty „za tečkou“; toto signalizuje, že se jedná o „**méně podstatná změnu**“,
- **změna verze:** změna atributu vyžaduje novou verzi instance objektu. V atributu číslo verze a varianty dochází tedy ke změně hodnoty „před tečkou“; toto signalizuje, že se jedná o „**podstatnou změnu**“.

2.3.3 Objekt #ObjektZávislost

Účelem tohoto objektu je možnost definovat závislosti mezi reálně existujícími objekty systému SDAT. Každá definice závislosti se skládá z těchto částí:

- určení primárního objektu, tj. objektu, u kterého se sleduje, zda došlo ke změně,
- určení závislého objektu, tj. objektu, který vyžaduje automatické vytvoření nové instance v případě, že došlo k vytvoření nové instance primárního objektu,
- určení druhu změny u primárního objektu (atribut „změna primárního“),

[1] Pro toto pravidlo existuje výjimka: atribut, který je označen jako „změna hodnoty není povolena“, je měnitelný v případě, že k aktuálnímu datu není instance objektu platná a (v případě, že jsou u objektu sledovány stavy) zároveň je instance objektu ve stavu Projektovaný.

- určení toho, jaký druh změny vyvolává definovaná změna instance primárního objektu u instance objektu závislého (atribut „změna závislého“),
- určení toho, pro jaký stav primárního objektu toto nastavení platí.

Vazby mezi objekty se definují vždy pro nejbližší objekty podle objektového modelu. Nemůže tak například být definována vazba mezi objektem Datová oblast a Výkaz, protože podle objektového modelu stojí mezi těmito objekty objekt Blok výkazu.

Na objektech Datová oblast, Blok Výkazu a Výkaz lze výše uvedená pravidla převést do podoby příkladu. Byznys požadavek zní tak, že v případě, kdy dojde k vytvoření nové verze Datové oblasti, musí dojít k vytvoření nové verze Výkazu. Aby bylo možno dosáhnout výše uvedeného chování, pak je třeba splnit následující podmínky:

- Objekt #Objekt musí obsahovat tyto instance:
  - Výkaz
  - Blok Výkazu
  - Datová oblast
- V rámci objektu #ObjektZávislost budou nadefinovány dvě instance:
  - Instance č. 1
    - Primární objekt: Datová oblast
    - Závislý objekt: Blok Výkazu
    - Změna primárního: Nová verze
    - Změna závislého: Nová verze
    - Stav primárního: Platný
  - Instance č. 2
    - Primární objekt: Blok výkazu
    - Závislý objekt: Výkaz
    - Změna primárního: Nová verze
    - Změna závislého: Nová verze
    - Stav primárního: Platný

Pomocí výše uvedeného nastavení jsme dosáhli toho, že v případě, kdy uživatel vytvoří novou verzi objektu Datová oblast tak, že začne upravovat Datovou oblast, která je platná, pak systém:

- musí vytvořit novou verzi související instanci objektu Blok výkazu,
- jakmile je vytvořena nová verze instance objektu Blok výkazu, která vznikla z jiného Bloku výkazu ve stavu Platný, pak musí vzniknout nová verze Výkazu.

Takto definované závislosti umožňují spustit proces tzv. **nuceného verzování**. K procesu nuceného verzování dojde v okamžiku, kdy jsou splněny tyto podmínky:

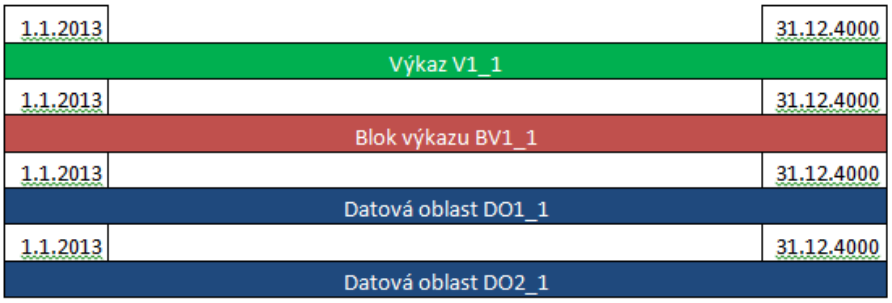
- je vytvářena verze/varianta instance nějakého objektu, který má zároveň nějakou nadřízenou instanci objektu,
- pro oba objekty existuje definice primární/závislý objekt (existuje instance objektu #ObjektZávislost).

Definice závislostí postupuje „**od spodu nahoru“** (z hlediska objektového modelu to znamená, že postupuje od podřízených objektů k objektům nadřízeným). To znamená, že definuje, co se stane se souvisejícími nadřízenými instancemi objektů, pokud dojde ke změně instance objektu podřízeného. Definice závislostí mezi objekty bude využita tehdy, pokud je

prováděna změna instance nějakého objektu, která se váže k nadřízenému objektu. Typicky toto bude využito tehdy, pokud někdo v rámci existující verze Výkazu začne editovat Datovou oblast.

Následující obrázky jsou ilustrativní příklad pro zachycení principu verzování nadřízených a podřízených objektů „od spodu nahoru“. Pro jednoduchost je příklad rozpracován pouze do třetí úrovně vztahů objektu Výkaz:

Výkaz V1 je tvořen Blokem výkazu BV1, který obsahuje dvě Datové oblasti DO1 a DO2. Výchozí situace znamená, že všechny objekty jsou v první verzi (001.000).



Obrázek 3 - Verzování "od spodu nahoru" – výchozí situace

Od 1. 1. 2014 dochází ke změně Domény číselníku DM1, která je použita pro konkretizaci parametrů Údajů v Datové oblasti DO1. Předpokládejme, že vzniká verze Domény číselníku DM1 s číslem verze/varianty 002.000 s platností od 1. 1. 2014. Časová platnost předchozí verze Domény číselníku 001.000 DM1_1 bude ukončena k 31. 12. 2013 při převádění do stavu Platný. Na základě vzniku nové verze Domény číselníku 002.000 DM1_2 vznikají nové verze objektů Výkaz (002.000 V1_2), Blok výkazu (002.000 BV1_2) a dotčené Datové oblasti (002.000 DO1_2). Protože změna domény DM1 byla jedinou změnou definice Výkazu V1, není datové oblasti DO2 vytvářena nová verze v rámci verze Výkazu 002.000 V1_2.

Obrázek 4 - Verzování "od spodu nahoru" – výsledný stav

V případě, že uživatel zvolí opačný postup úpravy instancí objektů, tedy „**ze shora dolů**“ a editaci zahájí u instance objektu, která je z hlediska hierarchie objektů nejvýše, pak to automaticky znamená, že k nové verzi nebo variantě jsou automaticky přivázány všechny podřízené instance objektů ve stejné verzi/variantě a se stejnou platností, s jakou byly přivázány k instanci nadřízeného objektu před vytvořením nové verze. Tento přístup bude využit tehdy, pokud se uživatel rozhodne vytvořit novou verzi/variantu existující instance objektu Výkaz. V takovém případě, po vytvoření této nové verze/varianty, budou k této nové

verzi/variantě přivázány všechny instance objektu Blok Výkazu (a jemu podřízené) v takové verzi/variantě a s takovou platností, s jakou byly přivázány k instanci nadřízeného objektu, ze které vznikla nová verze/varianta instance objektu.

Aby tento aparát fungoval, musejí být splněna následující pravidla:

a) kombinace objektů primární objekt/závislý objekt může existovat maximálně jednou a to i v opačném pořadí (pokud jsou tedy dva objekty ve vztahu primární/závislý, nelze pro tytéž objekty definovat reverzní vztah),
b) není možné, aby, pokud existuje vazba primární objekt/závislý objekt, existovalo obrácené pořadí této vazby, tedy závislý objekt/primární objekt (to by vedlo k cyklické závislosti). Toto pravidlo platí v celé kaskádě závislostí. Nesmí se tak stát, že by změna A vedla ke změně B, změna B ke změně C a změna C ke změně A,
c) jeden Primární objekt může mít N Závislých objektů,
d) konkrétní závislý objekt může být použit i jako primární, ale nikdy ne v kombinaci s objektem, pro který je definován jako závislý (viz bod b); vznikla by cyklická závislost).

2.3.4 Dopad editace instance objektu na číslo verze a varianty

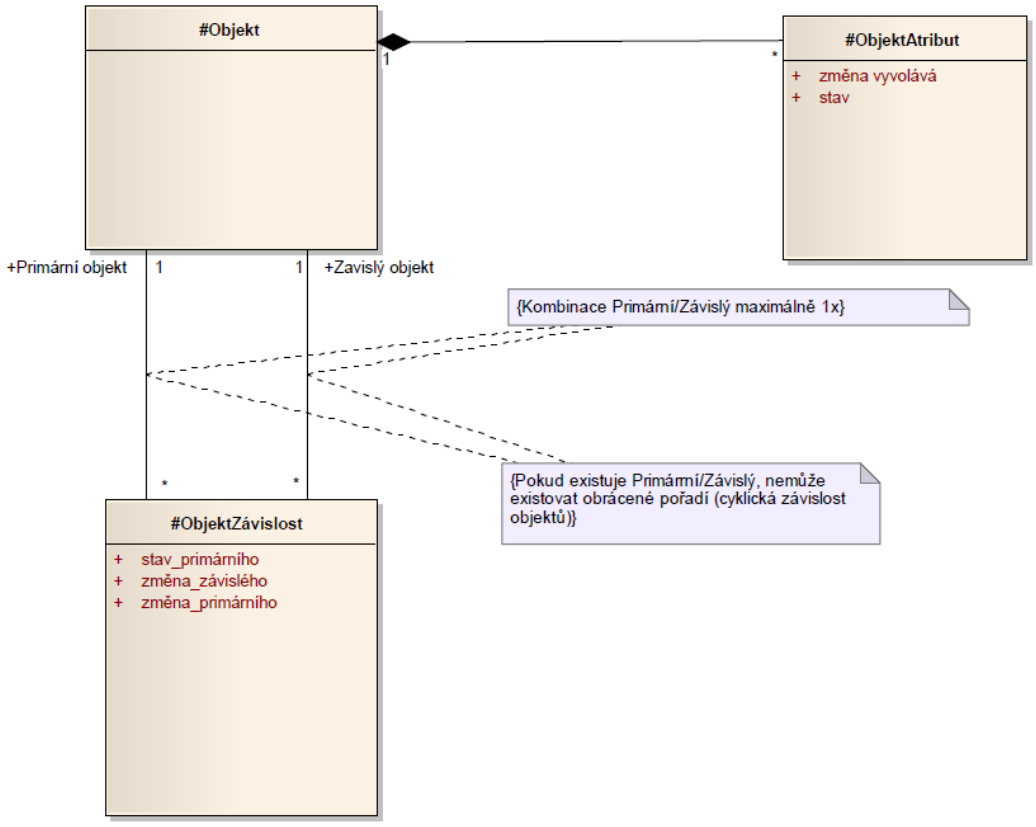
V případě, že se uživatel rozhodne měnit nějakou existující instanci jakéhokoli objektu metapopisu, postupuje se takto:

1. Uživatel vybere konkrétní instanci objektu, kterou chce měnit. V případě objektů, které sledují historii, se jedná o instanci, která je vymezená časovou platností (atributy platnost_od a platnost_do daného objektu).
2. Uživatel se rozhodne, jaký typ změny chce provést, k dispozici má volby:
    a. vytvořit novou verzi instance objektu,
    b. vytvořit novou variantu instance objektu,
    c. provést prostou editaci záznamu bez toho, že by vznikla nová verze nebo varianta objektu.
3. Systém nabídne uživateli atributy instance objektu, který si vybral v bodě 1. k editaci a uživatel jeden nebo více z nich změní.
4. Systém zjistí, zda provedené změny jsou povoleny v kontextu vybrané akce (změna verze, změna varianty, bez verze/varianty). Toto zjištění provede takto:
    a. systém zjistí, instance jakého objektu je modifikována, jaké atributy daného objektu byly změněny a případně v jakém stavu se instance objektu nachází (to pouze v případě, že se jedná o editaci instance objektu, který sleduje stavy),
    b. systém pro všechny změněné atributy dané instance z objektů #Objekt a #ObjektAtribut zjistí, jakou akci změna vyvolává (v případě, že se jedná o instanci objektu, který sleduje stavy, musí se do vyhodnocení zohlednit i stav instance objektu),
    c. pokud systém zjistí, že změny provedené uživatelem jsou v kolizi s nastavením systému (například se uživatel rozhodl, že vytvoří novou variantu instance objektu, ale změnil hodnotu atributu, který smí být měněn jen v rámci nové verze), systém danou změnu neprovede. Systém uživateli oznámí, že danou změnu nemůže provést, protože je v kolizi s nastavením systému a nabídne uživateli vytvoření verze (místo varianty) a vyžádá si zadání všech dodatečných informací, které jsou pro vytvoření nové verze třeba,

   d. v případě, že změny provedené uživatelem v kolizi s nastavením systému nejsou (za kolizi se nepovažuje, pokud je povolená akce „vyšší“ než druh provedené změny. Například kolizí není, pokud uživatel určí, že chce vytvořit novou variantu a změní hodnotu atributu, pro který je nastaveno, že jej lze měnit bez vytváření verze nebo varianty), pak systém provede uživatelem vybranou akci, tedy:
      i. vytvoří novou verzi instance objektu,
      ii. vytvoří novou variantu instance objektu,
      iii. provede editaci záznamu (update) bez toho, že by vytvářel jakoukoli novou instanci objektu (existující hodnota v měněných atributech bude přepsána hodnotou novou, stará hodnota bude ztracena).

5. V případě, že systém vytvoří novou verzi nebo variantu dané instance, systém musí provést kontrolu souvisejících instancí objektů takto:
   a. v případě, že je objekt, jehož instance byla změněna, definován jako primární objekt, systém musí provést stejnou akci jako s instancí primárního objektu také s instancí závislého objektu. Pokud v rámci primárního objektu vznikne nová verze nebo varianta, pak systém zkontroluje, zda existuje definice závislostí pro danou změnu (atribut „změna primárního“) a pokud ano, pak musí provést v rámci závislého objektu akci definovanou v atributu „změna závislého“,
   b. kontrola a činnost systému popsaná v bodě a. musí proběhnout kaskádně. To znamená, že pokud nová verze instance objektu A vyvolala vytvoření nové verze objektu B (protože A je definován jako primární a B jako závislý objekt), pak musí vzniknout i nová verze instance objektu C (protože B je definován jako primární a C jako závislý objekt).

2.3.5 Objektový model



Obrázek 5 - Objektový model – Definice závislostí mezi objekty

2.3.6 Rámcové vymezení závislosti objektů

Při vytvoření nové verze/varianty objektu nebo vazeb mezi objekty může dojít k vytvoření více nových verzí jednoho nebo více typů objektů v závislosti na jeho použití a ke kaskádovitému verzování objektů v závislosti na tom, jak jsou jednotlivé objekty provázány. Následující tabulka uvádí příklady možných změn bez jejich kaskádovitého vlivu, který lze odvodit. Přesný výčet změn je součástí podrobné analýzy dodavatelem.

Změna	Vyvolaná změna
Nová verze Bloku	• Nová verze Výkazu
Nová varianta Bloku	• Nová varianta Výkazu
Nová verze Datové oblasti	• Nová verze Bloku
Nová varianta Datové oblasti	• Nová varianta Bloku
Nová verze Hierarchie číselníku	• Nová verze Datové oblasti, v jejíž konkretizaci Parametru je použita • Nová verze kontroly (JVK, MVK)

Změna	Vyvolaná změna
Změna obsahu Hierarchie číselníku	• Nová verze příslušných Domén číselníku, které jsou modifikovány • Nová verze příslušných Hierarchií číselníků, které jsou při synchronizaci modifikovány
Nová verze Domény číselníku	• Nová verze Datové oblasti, v jejíž konkretizaci Parametru je použita • Nová verze kontroly (JVK, MVK)
Nová verze Parametru	• Nová verze Datové oblasti, v jejíž konkretizaci je Parametr • Nová verze Datové oblasti, v jehož konkretizaci Ukazatele je použita (Dodatečná konkretizace ukazatele)
Nová varianta Parametru	• Nová varianta Datové oblasti, v jejíž konkretizaci je Parametr • Nová varianta Datové oblasti, v jejíž konkretizaci Ukazatele je použita (Dodatečná konkretizace ukazatele)
Nová verze Datového typu	• Nová verze Datové oblasti, v jejíž konkretizaci Parametru je použita • Nová verze Kontroly
Nová verze Domény datového typu	• Nová verze Datové oblasti, v jejíž konkretizaci Parametru je použita • Nová verze Kontroly • Nová verze Datové oblasti, v jejíž konkretizaci Ukazatele je použita (Dodatečná konkretizace ukazatele)
Nová verze Ukazatele	• Nová verze Datové oblasti
Nová varianta Ukazatele	• Nová varianta Datové oblasti
Založení nového Účtu (tj. změna obsahu Účtové osnovy)	• Nová verze Ukazatele, je-li vazba Ukazatele na Účtovou osnovu definována celou skupinou nebo třídou, do níž byl nový Účet přidán
Ukončení platnosti Účtu (tj. změna obsahu Účtové osnovy)	• Nová verze Ukazatele, kde byl Účet přiřazen
Nová verze/varianta kontroly	• Nová varianta Výkazu

Tabulka 1 – Změny verzí a variant v důsledku závislostí objektů

K těmto změnám (nové verze/varianty) dochází v příslušném časovém řezu, pokud ještě nová verze/varianta není vytvořena.

V případě, že je vytvořena nová varianta nadřízeného objektu a změna podřízeného objektu vyžaduje novou verzi nadřízeného objektu, dochází k nahrazení stávající varianty nadřízeného

objektu novou verzí (s odpovídajícím přečíslováním). Pokud změna vyžaduje novou variantu a je již vytvořena nová verze, nedochází ke změně.

Jeden objekt může být současně nadřízeným i podřízeným, záleží na tom, jak je objekt použit (např. Doména číselníku, která je použita při definování obsahu dimenzionálního Parametru, je v tomto případě podřízeným objektem. V jiném případě, kdy Doména číselníku je definována Položkami číselníku, je pak Doména nadřízeným objektem).

Pro vzájemné používání objektů z hlediska stavu verze nebo varianty objektu platí následující:

Nadřízený objekt	Podřízený objekt		
	Projektovaný	Schválený	Platný
Projektovaný	Ano	Ano	Ano
Schválený	Ne	Ano	Ano
Platný	Ne	Ne	Ano

Tabulka 2 - Podřízené a nadřízené objekty

2.4 Objekty metapopisu

V této kapitole jsou uvedena obecná pravidla platící pro všechny objekty metapopisu.

2.4.1 Základní vlastnosti objektů

Každý objekt je nutno sledovat s ohledem:

- na časovou platnost objektu (atributy platnost_od a platnost_do). Časové platnosti musí být v rámci časové osy jednoznačné a nesmí se překrývat u objektů, které jsou ve stavu Platný. Pro poslední verzi nebo variantu (neplatí pro ukončené objekty) je platnost_do stanovena fixním datem v budoucnosti (např. 31. 12. 4000),
- na obsah změn (viz kapitola 2.3 Vazby mezi jednotlivými objekty), který se promítá do:
  - verze objektu, která je použita tehdy, pokud se jedná o podstatnou změnu, která ovlivňuje Hodnotu údaje,
  - varianty objektu, která představuje zachycení méně podstatné změny, jež je pouhým upřesněním některého z atributů objektu, které věcně nemění daný objekt. Varianta objektu je sledována v rámci verze objektu,
  - bez zachycení, tj. nepodstatné změny jako opravy překlepů a dalších formálních změn, které není potřeba promítat do nové verze ani varianty. Systém vede pouze evidenci těchto změn bez číslování v rámci verzí nebo variant (používá se při prezentaci metapopisu – viz kapitola 5.7 Proces prezentace Výkazu),
- na použití objektu, kdy se zobrazuje použití podřízeného objektu ve všech nadřízených objektech ve zvoleném časovém kontextu. Uživatel volí časový řez a má možnost zobrazit použití v tomto časovém řezu nebo v budoucím období, tj. v nadřízených instancích objektů, kterých platnost_od je datum v budoucnosti. Rozsah a algoritmy jsou součástí detailní analýzy dodavatelem.

2.4.2 Standardní atributy objektů

V dalším textu budou definovány jednotlivé objekty metapopisu a vazby mezi nimi. V této kapitole jsou uvedeny standardní atributy objektů, tedy atributy, které jsou součástí každého z objektů metapopisu, pokud není u daného objektu explicitně uvedeno, že tomu tak není. Standardní atributy proto už nejsou u jednotlivých objektů uváděny, uváděna je pouze odchylka od standardních atributů.

V závislosti na tom, zda a jak daný objekt metapopisu podléhá sledování historie, případně historie a stavů, jsou standardní atributy umísťovány do jednotlivých objektů. Objekt, který nijak nesleduje historii má všechny standardní atributy v hlavním objektu (_Objekt), naopak objekt, kde se historie sleduje, má většinu standardních atributů na objektu _ObjektHistorie (s výjimkou atributu Kód, který je v čase neměnný a je vždy na hlavním objektu (_Objekt).

Každý z objektů metapopisu může mít definovány ještě další atributy, a to nad rámec zde uvedených standardní atributů.

Níže je uvedena základní charakteristika a způsob práce s jednotlivými standardními atributy, detailní specifikace a popis způsobu práce s atributy konkrétních objektů je popsán v kapitole 7 Funkční požadavky.

Standardní atributy jsou:

a) Interní identifikátor objektu

Interní identifikátor objektu je jednoznačný systémový interní identifikátor, který je objektu přidělen systémem při založení objektu, po dobu životnosti objektu se nemění a při ukončení platnosti objektu nemůže být přidělen jinému objektu. Uživatelům není prezentován. Je povinný.

b) Kód objektu

Kód objektu je jednoznačný identifikátor v rámci typu objektu, který je objektu přidělen při jeho založení. Kód objektu je mimo jiné uživatelem využíván k vyhledávání daného objektu, ke třídění a seskupování. Kód objektu je povinný.

Pro stanovení kódu objektu platí následující:

- kód objektu může obsahovat vybrané alfanumerické a speciální znaky (např. velká písmena, číslice a podtržítko),
- kód objektu se po dobu životnosti objektu nemění (kód objektu lze pouze změnit za podmínky, že existuje pouze první verze, a to ve stavu Projektovaný, která není nikde použita,
- kód objektu je jednoznačný, pro jednotlivé typy objektů jsou kontroly jednoznačnosti uvedeny v kapitole 7 Funkční požadavky,
- kód objektu až na výjimky uvedené u jednotlivých objektů přiřazuje uživatel, pro jednotlivé typy objektů existují metodické konvence těchto identifikátorů.

c) Název objektu

Výstižné slovní označení věcného obsahu objektu. Název objektu většinou přiřazuje uživatel při jeho založení. V průběhu životnosti objektu lze název objektu měnit; změnu lze provádět ve verzi nebo variantě ve stavu Projektovaný. Nepodstatné změny neměnící charakter lze provádět i ve stavu Schválený. Název objektu slouží mj. k vyhledávání

objektů. Evidence názvu je možná v tzv. primárním jazyce (čeština; povinná hodnota), tak i v jazyce sekundárním (angličtina; nepovinná hodnota).

d) Popis objektu

Slovní vymezení věcného obsahu daného objektu. V průběhu životnosti objektu lze definici měnit; změnu lze provádět ve verzi nebo variantě ve stavu Projektovaný. Nepodstatné změny neměnící charakter lze provádět i ve stavu Schválený a Platný. Evidence popisu je možná v tzv. primárním jazyce (čeština; nepovinná hodnota), tak i v jazyce sekundárním (angličtina; nepovinná hodnota).

e) Poznámka

Případné dodatečné informace k objektu, které nejsou uvedeny v popisu objektu. V průběhu životnosti objektu lze poznámku měnit; změnu lze provádět ve verzi nebo variantě ve stavu projektovaném. Nepodstatné změny neměnící charakter lze provádět i ve stavu schváleném a platném. Evidence poznámky je možná v tzv. primárním jazyce (čeština; nepovinná hodnota), tak i v jazyce sekundárním (angličtina; nepovinná hodnota).

f) Autor objektu

Jednoznačná identifikace uživatele, který vytvořil novou verzi nebo variantu objektu. Přiřazuje ho systém při založení nové verze nebo varianty objektu. Zapsána je identifikace uživatele, který danou akci provedl. Atribut se v průběhu životnosti nemění, uživatel si tuto informaci může zobrazit.

g) Datum vytvoření

Datum a čas vytvoření nové verze nebo varianty přiřazuje systém při založení verze nebo varianty podle aktuálního data. Toto datum je v systému uloženo od okamžiku vzniku záznamu a nemění se po celou dobu životnosti tohoto objektu.

h) Kdo aktualizoval

Identifikace uživatele, který naposledy změnil verzi nebo variantu objektu. Je nastavován systémem automaticky.

i) Datum a čas aktualizace

Datum a čas poslední aktualizace instance objektu. Je nastavován systém automaticky. V průběhu životnosti objektu se mění, pokud dojde k přepsání záznamu.

j) Platnost_od

Datum, které vyjadřuje skutečné období, od kterého verze nebo varianta objektu bude platit, resp. platí. Toto datum nastavuje systém nebo uživatel, a to buď hromadně prostřednictvím platnostiMetodiky vykazovacího rámce, nebo individuálně v případě, že platnost objektů je rozdílná od platnosti celé Metodiky vykazovacího rámce (v případě objektu Výkaz). Při zakládání objektů v Knihovně se postupuje obdobně. Platnost_od lze měnit v průběhu projektování při dodržení pravidel sladění jednotlivých verzí a variant souvisejících objektů.

k) Platnost_do

Datum, do kterého platí verze nebo varianta objektu. Pokud za určitou verzí nebo variantou následuje další verze nebo varianta, systém přiřazuje platnost_do tak, aby

období platnosti plynule za sebou následovala. Konec platnosti verze/varianty se automaticky nastavuje při změně stavu Schválený – Platný následující verze/varianty.

V případě ukončení platnosti objektu bez dalšího pokračování hodnotu platnost_do přiřazuje uživatel v rámci funkce Ukončení platnosti objektu. Poslední verze nebo varianta, pokud objektu není ukončena platnost, má hodnotu nastavenu na 31. 12. 4000.

l) Garant

Identifikace zaměstnance ČNB, který odpovídá za obsahovou správnost objektu. Defaultně je nastaven autor objektu, změnu provádí uživatel na základě výběru nabídky z listu (vazba na seznam zaměstnanců ČNB).

2.4.3 Základní principy práce s objekty

Základní předpoklady pro práci s objekty v systému SDAT jsou především následující:

a) realizace všech aktivit při projektování je zajištěna bez aktuálního zapojení programátorských kapacit prostředky systému, tj. systém zajistí realizaci požadované akce okamžitě v systému,
b) všechny operace jsou prováděny z nabídky menu jednotlivých objektů,
c) změny stavů u objektů jsou prováděny automaticky systémem nebo uživatelem podle typu prováděné akce,
d) je vedena přehledná evidence všech objektů systému,
e) jsou umožněny hromadné akce v rámci práce s objekty,
f) existují interaktivní nápovědy u každého použitého objektu s uvedením postupu práce pro možné druhy činností s objektem,
g) je umožněno vyhledávání objektů podle výběrových kritérií odpovídajících atributů a vazeb daného objektu (viz dokument A – Obecné požadavky, kapitola 4.6.1 Komponenta Tabulka dat (grid)),
h) systém automaticky zaznamenává přechody stavů objektů (Projektovaný ->Schválený->Platný) a eviduje data přechodu a autora změny,
i) je možné vyhledat a zobrazit použití objektu v jeho nadřízených objektech s ohledem na časovou platnost:
    a. pro použití objektu v aktuálním časovém kontextu,
    b. pro časový kontext i do budoucnosti (např. pro dosud nezaložené verze neukončených objektů),
j) veškeré činnosti s objekty jsou zaznamenány a jsou viditelné všem uživatelům podle přiřazených přístupových práv,
k) je zajištěno logování všech akcí s objekty systémem s možností zobrazení např. přehledné historie změn u každého objektu a atributu.

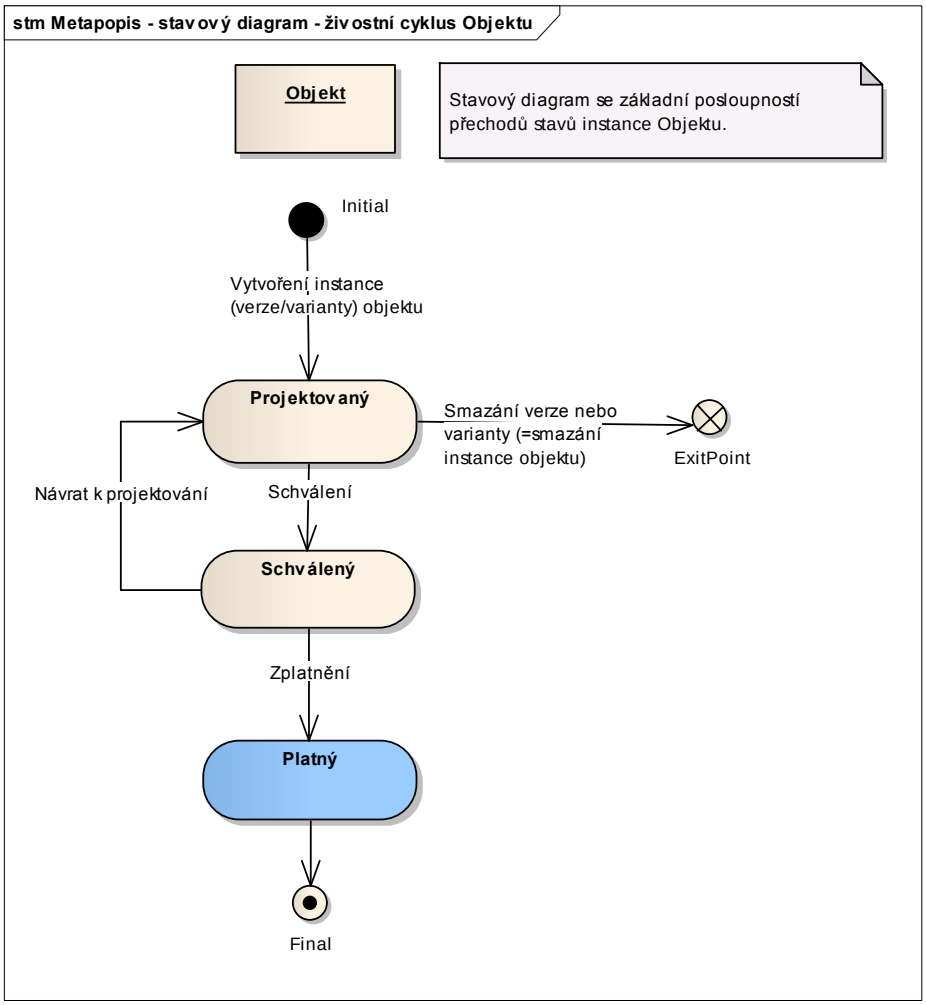
Pro všechny objekty platí základní pravidla pro práci s nimi, která se odvíjejí od stavu objektu, tj. musí platit následující:

Stav verze objektu	Definování rozsahu stavu	Posloupnost stavů	Možnost kroku zpátky	Možnost měnit objekt
Projektovaný	Od data vytvoření do data schválení	Projektovaný → Schválený → Platný	Ano = smazání nepoužité verze	Ano, všechny atributy mimo interní identifikátor objektu, typ objektu a označení verze/varianty
Schválený	Od data schválení do data platnost_od objektu	Schválený → Platný	Ano = do stavu Projektovaný	Ano, pouze atributy neměnící charakter objektu
Platný	Od data platnost_od objektu neomezeno	Platný	Ne = nutné vytvořit novou verzi	Ano, pouze atributy neměnící charakter objektu

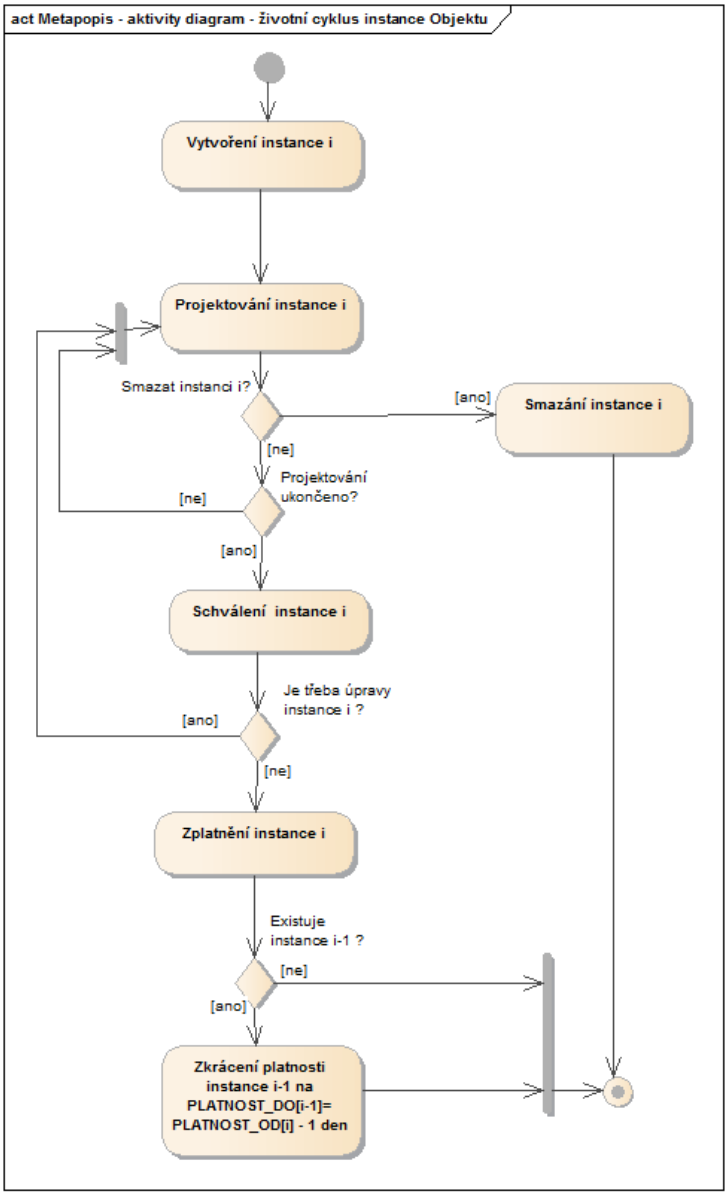
Tabulka 3 - Stavy objektu

2.4.4 Životní cyklus objektu

Každý objekt prochází stejným životním cyklem od jeho založení až do ukončení jeho platnosti nebo smazání. Organizace práce při přípravě výkaznictví vyžaduje, aby připravovaný systém minimálně v rámci životního cyklu objektu sledoval, resp. umožňoval následující přechody mezi níže vyznačenými stavy:



Obrázek 6 - Stavy objektů metapopisu - Stavový diagram



Obrázek 7 – Životní cyklus objektu Metapopisu – Aktivity diagram

V systému SDAT rozdělujeme datumovou platnost na tyto kategorie:

- **Odložená (metapopisná) datumová platnost** – v některých případech je možné vymezovat platnost nějaké entity systému tzv. odloženě – to znamená, že uživatel může například provést zplatnění výkazu 15. 12. 2016 s tím, že výkaz se stává platným 1. 1. 2017 (1. 1. 2017 je pak tzv. odložená (metapopisná) platnost, což znamená, že na období před tímto datem nelze pro daný výkaz vygenerovat výskyt výkazu, protože v tomto období výkaz ještě neplatí a 15. 12. 2016 je informace o tom, kdy reálně byla akce zplatnění provedena). Odložená datumová platnost bude nejčastěji použita v případě definice metapopisu, proto ji také můžeme nazývat „metapopisnou“ datumovou platností. Vyznačuje se tím, že systém uživateli umožňuje, aby sám nastavil časový úsek, po který nějaká instance objektu platí.

- **Reálná datumová platnost** – vymezuje časový úsek, po který bylo něco v systému reálně existující. Takové datumové platnosti generuje systém a vycházejí vždy z reálného časového okamžiku provedení nějaké akce a vyznačuje se tím, že uživatel nemůže rozhodnout o tom, od kdy do kdy reálně něco existuje – tato platnost je tak odvozena z chování uživatele v aplikaci (resp. provádění konkrétních akcí). Typickým příkladem takové reálné datumové platnosti je pak zařazení výkazu do Vykazovacího rámce. Datum_od tohoto zařazení je systémem stanoven na den (hodinu/minutu/sekundu), kdy uživatel zařadí výkaz do Vykazovacího rámce a datum_do je nastaveno na den (hodinu/minutu/sekundu), kdy uživatel reálně provede přeřazení výkazu do jiného Vykazovacího rámce. **Reálnou datumovou platnost nemůže uživatel sám změnit (změna možná pouze systémem při určité akci.**

Pokud nějaký objekt podléhá sledování historie + stavů 2.2.6 Přístup „Sledování historie – časová platnost + stavy“, pak pro vymezení rozsahu platnosti jednotlivých stavů platí:

- Stav „projektovaný“ je definován reálnou datumovou platností. Stav vznikne v okamžiku vzniku nové verze/varianty výkazu nebo uživatel převede výkaz ze stavu „schválený“ zpět do stavu „projektovaný“ (datum_od je nastaveno na datum, kdy uživatel reálně založí novou verzi/variantu, příp. změní stav zpět na „projektovaný“ ze stavu „schválený“) a zanikne ke dni (hodině/minutě/sekundě), kdy vznikne stav následný, v tomto případě stav „schválený“.
- Stav „Schválený“ je definován reálnou datumovou platností. Stav vznikne v okamžiku, kdy uživatel prohlásí entitu za schválenou. Není tedy možné žádnou z entit systému schvalovat v tzv. odloženém režimu, tedy „dnes říci, že entita se stane schválenou za týden“. Akt schválení je okamžitý a datum_od tak musí reflektovat systémové (reálné) datum provedení akce.
- Stav „Platný“ je definován odloženou datumovou platností. Stav sice vznikne v okamžiku, kdy uživatel prohlásí entitu za zplatněnou, nicméně jako datum_od je nastaveno „metapopisné datum“, které je evidováno u entity jako takové. U stavu „platný“ tak naopak platí, že okamžikem zplatnění neumožním Osobám vykazovat (musí si počkat na metapopisné datum_od) zároveň však ale změnou stavu ze „schválený“ na „platný“ znemožním např. návrat do stavu „projektovaný“.

2.4.5 Základní pravidla práce s instancemi objektů

Pro práci s objekty platí následující základní pravidla, specifická pravidla jsou uvedena u jednotlivých typů objektů.

2.4.5.1 Zakládání instance objektu

Pro založení instance objektu platí následující:

- založení instance objektu znamená založení jeho první verze a zadání minimálně povinných atributů objektu,

- instance objektu je zakládána uživatelem,
- objekt je zakládán z nabídky menu a typ objektu je automaticky přiřazen systémem při jeho založení,
- při založení systém eviduje autora a datum založení objektu,
- systém podporuje založení vybraných objektů importem z přesně definovaného rozhraní (např. Číselník lze založit hromadným natažením ze souboru, který bude odpovídat předdefinované struktuře rozhraní, dalším příkladem je objekt Ukazatel nebo objekt Kontrola),
- založení první verze objektu lze provést také replikací stávajícího objektu, kdy je kopírován původní objekt do nového objektu, který obsahuje stejné podřízené objekty jako původní. Novému nadřízenému objektu je systémem přiřazen nový interní identifikátor a uživatelem nový kód objektu,
- založení objektu je dostupné z Knihovny nebo z Metodiky vykazovacího rámce (viz kapitola 4 Knihovna).

2.4.5.2 Modifikace objektu

Pro modifikaci objektu platí následující:

- možnost provádět změny jednotlivých atributů objektů nebo měnit jejich obsah v jednotlivých stavech (Projektovaný, Schválený, Platný) a časových řezech verzí/variant objektů je řízena nastavením vzájemných závislostí objektů (viz kapitola 2.3.3 Objekt #ObjektZávislost). Toto nastavení lze uživatelsky změnit. Nastavení je jednotné pro celý systém a z pohledu jeho provozu je to zásadní nastavení (viz kapitola 2.3.4 Dopad editace instance objektu na číslo verze a varianty),
- systém sleduje, které atributy byly změněny, aby je mohl uvést v prezentaci změnového metapopisu,
- změny jsou prováděny z uživatelského rozhraní a systém automaticky eviduje autora a datum a čas (v případě založení nového záznamu atributy autor objektu a datum vytvoření, v případě editace atributy kdo aktualizoval a datum a čas aktualizace),
- provádění změn objektů je dostupné z Knihovny nebo z Metodiky vykazovacího rámce,

2.4.5.3 Ukončování platnosti objektů.

- Pro ukončování platnosti objektů platí následující:
- ukončením platnosti se rozumí ukončení poslední platné verze nebo varianty objektu nastavením atributu platnost_do na požadované datum. Platnost objektu je možno prodloužit (viz kapitola 2.4.5.5 Prodloužení platnosti objektu),
- nelze ukončit platnost objektu z Knihovny, který je použit v nadřízených objektech,
- pokud je ukončena platnost nadřízených objektů, může být současně na pokyn uživatele hromadně selektivně ukončena platnost podřízených objektů za předpokladu, že nejsou použity v dalších nadřízených objektech (závislosti objektů při ukončování jejich platnosti budou rozpracovány v rámci analýzy). Například s ukončením platnosti Výkazu se ukončí platnost i jeho Blokům výkazu a Datovým oblastem. Ukazatelům a Doménám číselníku, které jsou v ukončované instanci objektu Výkaz použity, se ukončí platnost pouze za předpokladu, že nejsou použity v jiných Výkazech a uživatel to požaduje. Dalším

typickým příkladem je, že s ukončením platnosti Datové oblasti (v nové verzi Výkazu) lze ukončit platnost např. Ukazatelům. Tento postup má za účel úsporu pracnosti při provádění změn ve výkaznictví. Je na rozhodnutí uživatele, zda zvolí postup pracnější (objekt po objektu) nebo hromadnou funkci,
- ukončování platnosti objektů je dostupné z Knihovny nebo z Metodiky vykazovacího rámce,
- ukončovat platnost objektu je možno pouze ve stavu Platný.

2.4.5.4 Smazání objektů

Pro smazání objektu platí následující:

- smazáním objektu se rozumí odstranění poslední nepoužité verze objektu (za předpokladu konzistence objektů, jinak systém smazání nepovolí),
- smazat je možné objekt ve stavu Projektovaný,
- v případě smazání objektu v jeho první a nepoužité verzi lze opětovně použít i jeho kód,
- smazání objektů je dostupné z Knihovny nebo z Metodiky vykazovacího rámce.

2.4.5.5 Prodloužení platnosti objektu

Pro prodloužení platnosti objektu platí následující:

- prodloužení platnosti objektu je možné provést pro objekty, které mají ukončenu časovou platnost,
- prodloužení platnosti objektu je možné za dodržení nastavených podmínek konzistence, např. lze prodloužit platnost poslední verze, jíž byla ukončena platnost tak, aby nebyla narušena časová návaznost verzí nebo variant použitých objektů,
- konec platnosti poslední verze/varianty objektu při Prodloužení platnosti objektu je defaultně nastaven na 31. 12. 4000

2.5 Kontrola konzistence

Účelem kontroly konzistence je zejména zjistit, jaký je stav projektování a zda je metapopis za vybranou oblast výkaznictví hotov a je možno převést objekty do stavu Schválený nebo Platný. Kontrolu konzistence lze spustit i pro Výkazy ve stavu Platný. Kontrola konzistence zahrnuje jednotlivé kroky, které mají stanovený následující stupně důležitosti:

a) **Chybný:** jedná se o závažnou chybu, která neumožňuje převést objekty metapopisu do stavu Schválený nebo Platný. Pokud je nalezen alespoň jeden nesplněný krok se stupněm důležitosti Chybný, všechny objekty metapopisu, u nichž nebyly splněny kontroly, a jim nadřízené objekty nejsou převedeny do stavu Schválený nebo Platný,
b) **Informativní:** je určen pouze pro informaci a je na uživateli, zda toto upozornění akceptuje a změní metapopis. Tento stupeň důležitosti nebrání převedení objektu do stavu Schválený nebo Platný.

Kontrola konzistence se provádí:

- vždy s určením data, ke kterému se bude kontrola provádět. K tomuto datu budou vyhledávány platné verze souvisejících objektů,
- automaticky v rámci spuštění funkce pro převedení objektů metapopisu do stavu Schválený nebo Platný.
- na požádání uživatele kdykoliv.

Kontrola konzistence zahrnuje zejména následující kroky:

- stupeň důležitosti Chybný:
  1. úplnost a správnost metapopisu:
     a. popis všech vykazovaných buněk (Údajů) ve Výkazu je dokončen,
     b. jsou správně definovány Datové oblasti (údaje mají shodné dimenzionální Parametry),
     c. neexistuje duplicita vykazovaných Údajů v rámci Výkazu,
     d. Ukazatele použité ve Výkazu mají přiřazen Datový typ,
     e. definice věcných kontrol je dokončena (sémantické kontroly mají uživatelský tvar, u algoritmických je programovací kód validní),
  2. soulad jednotlivých verzí/variant objektů, použitých pro metodický popis výkazu. Musí být splněny následující podmínky:
     a. platnost_od Výkazu leží v intervalu platnost_od a platnost_do objektů použitých pro popis Výkazu,
     b. pokud předchozí podmínku splňuje více instancí použitých objektů (toto může nastat pouze v případě, že daný objekt má verzi/variantu ve stavu Projektovaný nebo Schválený, ale ne ve stavu Platný, nebo je předchozí verzi/variantě zkrácena platnost_do), vezme se ta instance objektu, jejíž platnost_od je nejvyšší a současně rovna nebo menší než platnost_od Výkazu),
  3. kontrola navázání časových řad, tj. časová řada je jednoznačná a intervaly platnost_od a platnost_do verzí/variant Výkazů následníka a předchůdce se nepřekrývají (viz kapitola 6.2 Proces Navazování časových řad Údajů). Provádí se pouze, pokud jsou časové řady navazovány (nepovinné),
  4. pokud výkaz obsahuje sdílenou datovou oblast, musí být mateřská datová oblast schválena nebo zplatněna současně nebo dříve,
  5. existence použitých nesynchronizovaných Hierarchií číselníku,
- stupeň důležitosti „Informativní“
  1. objekty obsahující prázdné pole Popis (podle typů objektů),
  2. existence duplicitních Domén číselníků (domény se shodným obsahem Položek číselníku),
  3. existence Domén číselníku vytvořených současně z Číselníku a Hierarchie číselníku.
  4. výkazy se shodným atributem platnost_od používají shodné verze jednotlivých objektů.

Předpokládá se, že z analýzy, testovacího a běžného provozu vyplyne potřeba dalších kroků kontroly konzistence nebo jejich úprava.

Uživatel před spuštěním kontroly konzistence může vybrat jednotlivé kroky a rozsah (Metodika vykazovacího rámce, Výkazy, Datové oblasti, objekty Knihovny), pro který má být kontrola konzistence provedena. V rámci převádění objektů do stavu Schválený a Platný se provádějí vždy všechny kroky kontroly konzistence.

Systém po provedení kontroly konzistence v požadovaném rozsahu vypíše všechny objekty, které nesplňují požadované kroky kontroly konzistence, s jejich přesnou identifikací a stupněm důležitosti. Výsledky kontroly konzistence jsou uloženy v systému, uživatelé si je mohou zobrazit a vytisknout. Systém umožňuje výsledky kontroly smazat po uplynutí 3 let od jejich spuštění.

3 Objektový model

Tato kapitola obsahuje popis jednotlivých objektů metapopisu. Pro přiblížení počtu instancí níže definovaných objektů metapopisu slouží 8.4 říloha 4 — Počty instancí vybraných objektů metapopisu v roce 2016 v systému MtS.

3.1 Objekt Vykazovací rámec

Sledování historie objektu: Bez sledování historie (viz kapitola 2.2.3 Přístup „Bez sledování historie“)

Objekt „Vykazovací rámec“ je koncipován jako objekt nejvyšší úrovně a jeho účelem je základní rozčlenění výkazů v systému. Jedná se o jednoúrovňový číselník, jehož položky budou použity pro členění výkazů (předpokládá se členění dle sektorů, ale může se jednat o libovolné jiné kritérium). Každý vykazovací rámec má svůj jedinečný kód. Kromě základní navigace poskytuje tento objekt i základní objekt pro řízení přístupových práv (pomocí něj bude vznikat dynamická definice přístupových práv, více viz dokument „F – Uživatelé a přéstupová práva).

Každý Výkaz musí být v okamžiku založení, zařazen právě **do jednoho vykazovacího rámce** (jedná se příslušnost výkazu jako celku k vykazovacímu rámci), více viz 3.1.2 Objekt Výkaz ve Vykazovacím rámci.

Přímým podřízeným objektem k objektu „Vykazovací rámec“ je objekt „Metodika Vykazovacího rámce“.

3.1.1 Objekt Metodika vykazovacího rámce

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Ke každé instanci objektu „Vykazující rámec“ musí existovat alespoň jedna instance objektu „Metodika vykazovacího rámce“. Metodiky představují objekt, pomocí kterého se budou zastřešovat změny v jednotlivých výkazech (resp. verzích výkazu). Každá metodika v rámci vykazovacího rámce je označena kódem, který je v rámci daného vykazovacího rámce jedinečný. Předpokládá se, že kód metodiky vykazovacího rámce bude vytvořen z kódu vykazovacího rámce (do kterého metodika vykazovacího rámce patří) a bude přidán řetězec znaků, který bude tvořen datem, od kdy nabývají platnost změny ve verzích výkazů, které jsou do metodiky zařazeny.

Každá verze výkazu musí být zařazena **do právě jedné metodiky vykazovacího rámce**.

Důvodem pro udržování dvou vazeb (Výkazu na Vykazovací rámec a Verze výkazu na Metodiku vykazovacího rámce) je skutečnost, že objekt Vykazovací rámec bude sloužit k řízení přístupových práv (přístupová práva řídíme na výkazy, nikoli na jejich verze) a objekt Metodika vykazovacího rámce bude sloužit pro zastřešení změn ve výkazech, které se dějí prostřednictvím verzí výkazů.

Objekt Metodika vykazovacího rámce zajišťuje, že:

- Metodická platnost (platnost_od) verzí výkazu na sebe spojitě navazuje.
- Výkaz v Metodice vykazovacího rámce nemusí být nutně verzován, pokud v dané Metodice neprochází změnou.
- Metodická platnost (platnost_od) verze výkazu je dána platností/datem Metodiky vykazovacího rámce, ve které nabývají platnosti změny zaverzované výkazu; v případě, že v dané metodice není vytvořena nová verze výkazu, odvíjí se metodická platnost verze výkazu od atributu platnost_od Metodiky vykazovacího rámce, kdy byl Výkaz naposledy zaverzován.
- Metodická platnost (platnost_do) poslední verze výkazu, pokud nenásleduje další verze, nebo pokud výkazu nebyla platnost ukončena, je automaticky nastavena systémem na maximální datum.
- Metodické platnosti jednotlivých Metodik vykazovacího rámce na sebe musí spojitě navazovat.
- Výše zmíněná pravidla platí i pro další objekty (např. Blok výkazu, Datová oblast, apod.), které vznikly nebo byly zaverzovány v dané Metodice vykazovacího rámce.

Dále musí daný objekt umožňovat:

- Přiřazení verzí objektů z Knihovny potřebných pro projektování (např. Ukazatel, číselník, parametr, položka číselníku, hierarchie číselníku,…), resp. nabízení verzí potřebných objektů uživateli během projektování dle pravidla: metodická platnost_od verze objektu musí být shodná s platnost_od Metodiky vykazovacího rámce, nebo metodická platnost_od verze objektu musí být v intervalu, který obsahuje datum platnost_od Metodiky vykazovacího rámce.
- V případě potřeby uživatele ignorovat návrh vybrané verze objektu dle výše zmíněného pravidla a umožnit vybrat jinou věcně vhodnou verzi daného objektu. Úprava výběru oproti návrhu dle pravidla je možná pouze se speciální rolí na uživatele.
- Kontroly konzistence – viz kapitola 2.5 Kontrola konzistence.
- Ukončení platnosti Výkazu, resp. verzi Výkazu.
- Automatické navazování metodických platností verzí výkazu.
- Automatické navazování platností jednotlivých Metodik vykazovacího rámce v konkrétním Vykazovacím rámci.
- Možnost sdílení objektů mezi jednotlivými metodikami Vykazovacích rámců s zajištěním datové konzistence pro shodné platnosti,
- Prohledávat prostřednictvím nastavených kritérií a jejich kombinací (viz dokument A – Obecné požadavky, kapitola 4.6.1 Komponenta Tabulka dat (grid)),

- Provádět drobné úpravy na Výkazech ve stavu Schválený a Platný, které nemají vliv na obsah Údajů; tyto opravy nemají za následek vytváření nových verzí nebo variant objektů (viz kapitola 2.3 – Vazby mezi jednotlivými objekty), ale jsou evidovány systémem a v případě potřeby jsou publikovány.

Objekty Vykazovací rámec, Metodika vykazovacího rámce a další (viz níže), jejich vazby a procesy musí umožňovat:

- Technickou nezávislost mezi Metodikami vykazovacího rámce různých Vykazovacích rámců v případě, že neexistují jejich vzájemné závislosti (např. klonované DO, MVK apod.).
- Možnost ponechat rozpracovanou verzi výkazu (stav „Projektovaný“ nebo „Schválený) a danou verzi nezveřejňovat. Viz kapitola 3.1.3 Objekt Verze výkazu v Metodice.
- Možnost přesunu verze výkazu z Metodiky vykazovacího rámce do Metodiky jiného vykazovacího rámce s tím, že mezi verzemi výkazu musí být spojitá metodická platnost. Dále mezi dotčenými Metodikami vykazovacích rámců musí být spojitá metodická platnost. Přesouvat verze Výkazů lze pouze ve stavu „Projektovaný“ nebo „Schválený“. Dále lze přesouvat verze Výkazů pouze z Metodiky vykazovacího rámce, ve které aktuálně uživatelé projektují do jiné Metodiky vykazovacího rámce, ve které aktuálně uživatelé projektují.
- V případě věcné potřeby je možné tvořit kontroly (MVK) mezi verzemi výkazů v Metodikách Vykazovacího rámce různých Vykazovacích rámců.

Vykazovací rámec a Metodika vykazovacího rámce jsou zároveň používány jako hlavní navigační objekty pro strukturování metapopisu celého výkaznictví pro potřeby jeho prezentace ve webové aplikaci systému a pro potřeby předávání pomocí webových služeb.

3.1.2 Objekt Výkaz ve Vykazovacím rámci

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Informace o tom, **k jakému vykazovacímu rámci je výkaz přiřazen**, je podchycena v asociační třídě mezi objektem „Výkaz“ a „Vykazovací rámec“ s názvem „Výkaz ve vykazovacím rámci“. Součástí vazby výkazu na Vykazovací rámec je atribut „zařazen dne“, který zaznamená informaci o dni, kdy byl daný výkaz zařazen do vykazovacího rámce (atribut „vyřazen_dne“ bude při prvním zařazení výkazu nastaven na maximální datum). Systém umožňuje změnit zařazení výkazu do vykazovacího rámce; v takovém případě vznikne nová instance objektu „Výkaz ve Vykazovacím rámci“ s patřičnou hodnotou v atributu „zařazen_dne“ s tím, že u předcházejícího záznamu (kde je „vyřazen_dne“ nastaveno na maximální datum) se změní hodnota atributu na datum zařazení do nového vykazovacího rámce – 1 den.

V případě, že bude výkaz vyřazen z vykazovacího rámce, nemá to žádný vliv na to, že některé verze výkazů jsou zařazeny do metodik, které patří pod vykazovací rámec, z něhož je výkaz

vyřazen. Naopak ale platí, že v okamžiku vyřazení výkazu z vykazovacího rámce (a zařazení do jiného vykazovacího rámce) je nutno novou verzi výkazu zařadit pouze do takové metodiky, která je přivázána novému vykazovacímu rámci. Z toho plyne to, že v případě potřeby přeřadit výkaz do vykazovacího rámce musí automaticky dojít k vytvoření nové verze výkazu a daná verze výkazu v „novém“ vykazovacím rámci musí spojitě navazovat na metodickou platnost předchozí verze výkazu v původním vykazovacím rámci. Dále systém zajistí, aby nedošlo ke vzniku tzv. „datumové díry“, kdy datum zařazení do nového vykazovacího rámce bezprostředně nenevazuje na datum vyřazení z předchozího vykazovacího rámce.

3.1.3 Objekt Verze výkazu v Metodice vykazovacího rámce

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Informace o tom, **jaké verze výkazy jsou zařazeny do jakých metodik**, je podchycena v asociační třídě mezi objektem Výkaz (resp. VýkazVerze) a objektem „Metodika Vykazovacího rámce“ s názvem „Verze výkazu v Metodice“.

Součástí vazby je atribut „enabled“ (default „true“) s pomocí kterého je možno pozastavit distribuci Verze výkazu, pokud z nějakého důvodu je nutno uvolnit metodiku jako celek, ale některá z verzí výkazů není dokončená, nebo ji není možné z nějakého dalšího důvodu distribuovat a zároveň už ve verzi výkazu byly provedeny změny, o které není žádoucí do budoucna přijít. V takovém případě uživatel nastaví příznak „enabled“ na „false“ a systém zajistí, že sice bude umožněno takovou metodiku distribuovat, ovšem bez takto označené verze výkazu. Verze výkazu však zůstane v systému zachovaná.

Příznak „enabled“ na „false“ je možné nastavit pouze u verze Výkazu ve stavu Projektovaný nebo ve stavu Schválený. Verze Výkazu ve stavu Platný musí být vždy distribuovány.

Pokud verze Výkazu není distribuována, je poté přesunuta a doprojektována v následné Metodice vykazovacího rámce, platnost_od verze Výkazu se řídí atributem platnost_od Metodiky vykazovacího rámce.

3.1.4 Objekt Štítky výkazu

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Objekt Vykazovací rámec poskytuje možnost rozčlenit výkazy pouze podle jednoho kritéria a navíc vyžaduje, aby v jeden okamžik byl výkaz zařazen do právě jednoho vykazovacího rámce; to z hlediska členění výkazů podle různých hledisek není ideální konstrukt. Aby bylo možno tento nedostatek ošetřit, je zaveden objekt „Štítky výkazu“, který je koncipován jako jednoúrovňový číselník navzájem nezávislých nálepek (štítků), které se mohou libovolně přidávat k výkazům.

Zatímco číselník vykazovacích rámců musí být dopředu promyšlen a jeho obsah je víceméně konstantní v čase a tento obsah by (například při potřebě číselník vykazovacích rámců rozšiřovat) měl ctít jedno jediné předem vybrané kritérium, štítky mohou vznikat libovolně a hlavně mohou být koncipovány podle různých kritérií.

Štítky jsou vázány k Výkazu (nikoli k jeho verzi) vždy s vymezením časové platnosti navázání (platnost_od je datum kdy uživatel přilepí štítek k výkazu a platnost_do je datum, kdy štítek pro daný výkaz již neplatí, defaultně je toto datum nastaveno na maximální datum a ve většině případů se již nezmění). Jeden výkaz nemusí mít žádný, ale může mít N (a to dokonce v jednom čase) štítků. Platí, že jeden a ten samý štítek nesmí být k výkazu přiřazen v jeden čas více než jednou. Informace o navázání štítku na Výkaz je zachycena pomocí asociační třídy, která je mezi objekty Výkaz/Štítky výkazu s názvem „Označení výkazu štítkem“, a lze dle této informace Výkazy filtrovat.

3.2 Objekt Výkaz

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Objekt Výkaz sdružuje do jednoho logického celku údaje (přes Blok výkazu a Datovou oblast), které spolu logicky souvisejí z hlediska vykazování, a tvoří tak zásadní objekt sběru dat. Na objekt Výkaz je následně definována tzv. Vykazovací povinnost, resp. Výskyt výkazu, neboli předpis toho, jaká data, za jakou osobu, za jaké období a do jakého termínu mají být do ČNB předána.

Vztah objekt Výkaz k objektu Vykazovací rámec a Metodika vykazovacího rámce je popsán výše.

Dále platí, že Výkaz má vazbu na objekt Blok Výkazu, přičemž platí:

- Výkaz může vzniknout, aniž by obsahoval alespoň jeden Blok výkazu. Blok výkazu je k Výkazu přiřazen pomocí asociační třídy „Blok ve výkazu“; v rámci této třídy je třeba udržovat hodnotu atributu „pořadí“ (celé kladné číslo, které je v rámci Výkazu jedinečné), který udává pořadí bloku výkazu v rámci Výkazu. Pokud vzniká nový Výkaz, systém zajistí vytvoření Bloku včetně uvedení pořadí bloku ve výkazu.
- Výkaz může mít přiřazeno N (neomezeně) Bloků výkazů,
- pokud vzniká Blok výkazu, musí být zařazen k právě jednomu Výkazu,
- V případě potřeby změnit zařazení Bloku výkazu k Výkazu dojde k přepsání informace o tom, do jakého Výkazu je Blok výkazu zařazen (Blok výkazu „BL1“ má původně vazbu na Výkaz „V1“ a po změně má vazbu na „V2“, přičemž dojde ke ztrátě informace, že v minulosti měl Blok výkazu „BL1“ vazbu na Výkaz „V1“),
- pokud dojde ke smazání Výkazu, budou automaticky smazány všechny Bloky výkazu, které jsou k mazanému Výkazu připojeny (pozor, zde dojde ke kaskádnímu mazání, Blok výkazu obsahuje Datové oblasti a tyto Datové oblasti budou smazány stejně tak, viz další text).

Dále platí, že Výkaz má vazbu na objekt Kontrola, přičemž platí:

- Výkaz smí vzniknout, aniž by obsahoval alespoň jednu Kontrolu,
- Výkaz smí mít přiřazeno N (neomezeně) Kontrol,
- jedna Kontrola smí být přiřazena více Výkazům, nicméně právě v rámci jednoho Výkazu je označena jako „primární“ (je to ten Výkaz, v rámci kterého byla Kontrola vytvořena). Vazba na jiné výkazy je zřízena kvůli Mezivýkazovým kontrolám (MVK). Důvodem je požadavek, aby z Výkazu, na který je v rámci MVK odkazováno, bylo možno zjistit,

kterých všech kontrol se takový Výkaz účastní a zároveň byla kontrola modifikována na jednom místě. Tento požadavek je splněn pomocí asociační třídy „Připojení kontroly k Výkazu“,
- pokud dojde k smazání Výkazu, budou smazány všechny kontroly, které jsou k mazanému Výkazu připojeny, včetně všech vazeb na další Výkazy, které se účastní MVK. Související Výkazy jako takové nebudou smazáním primárního Výkazu nijak ovlivněny.

3.2.1 Atributy objektu Výkaz

Objekt Výkaz obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů).

Nad rámec standardních atributů obsahuje objekt Výkaz následující atributy:

- **konzistence (boolean):** určení toho, zda je Výkaz v rámci systému a své časové platnosti konzistentní. Atribut nabývá hodnoty „ano“, pokud Výkaz projde úspěšně kontrolou celkové konzistence. Hodnoty jsou nastavovány systémem a implicitně je atribut nastaven na hodnotu „ne“. Pokud je již konzistentní Výkaz modifikován, je atribut „konzistence“ nastaven zpětně na hodnotu „ne“ do doby než Výkaz znovu úspěšně projde kontrolou celkové konzistence,
- **vlastník dat:** identifikace organizačního útvaru ČNB (vazba na Řídící databázi ČNB), který je vlastníkem dat Výkazu. Vlastník dat (řídící pracovník útvaru) schvaluje oprávnění pro přístup k datům Výkazu.
- **synchronizovat na test:** řídící atribut synchronizace mezi produkčním a testovacím prostředím, podrobněji kapitola 2.1.2 Testovací prostředí dokumentu A – obecné požadavky.

3.3 Objekt Blok výkazu

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Účelem objektu Blok výkazu je rozdělit objekt Výkaz na menší logické celky. Blok výkazu je objekt, ze kterého je tvořen Výkaz. Vztah objektu Blok výkazu k objektu Výkaz je popsán výše.

Blok výkazu má vazbu na objekt Datová oblast, přičemž platí:

- Blok výkazu může vzniknout, aniž by obsahoval alespoň jednu Datovou oblast. Datová oblast je k bloku přiřazena pomocí asociační třídy „Datová oblast v bloku“; v rámci této třídy je třeba udržovat hodnotu atributu „pořadí“ (celé kladné číslo, které je v rámci Bloku výkazu jedinečné), který udává pořadí Datové oblastiv rámci Bloku. Pokud vzniká nový Blok, systém zajistí vytvoření Datové oblasti včetně uvedení pořadí datové oblasti v bloku výkazu.
- Blok výkazu může mít přiřazeno N (neomezeně) Datových oblastí,
- pokud vzniká Datová oblast, musí být zařazena k právě jednomu Bloku výkazu,
- V případě potřeby změnit zařazení Datové oblasti k Bloku výkazu, dojde k přepsání informace o tom, do jakého Bloku výkazu je Datová oblast zařazena (datová oblast „DO1“ má původně vazbu na Blok výkazu „BL1“ a po změně má vazbu na „BL2“,

přičemž dojde ke ztrátě informace, že v minulosti měla Datová oblast „DO1“ vazbu na Blok výkazu „BL1“),

- pokud dojde ke smazání Výkazu, budou automaticky smazány všechny související Bloky výkazu. Pokud dojde ke smazání Bloku výkazu, budou smazány všechny související Datové oblasti, které jsou do Bloku výkazu zařazeny (kaskádní mazání),

3.3.1 Atributy objektu Blok výkazu

Objekt Blok výkazu obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů) s výjimkou následujících:

- Garant

3.4 Objekt Datová oblast

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Účelem objektu Datová oblast je hromadně popsat ekonomické veličiny stejných vlastností, a to v rámci Bloku výkazu. Datová oblast je část Výkazu, která je tvořena Údaji. Tyto Údaje jsou následně popsány pomocí Ukazatelů a parametrů. Datová oblast tak představuje multidimenzionální datovou kostku, na jejíž osy (x, y, z) je možné podle návrhu uživatele umístit 1 až n dimenzí, tj. objektů typu Ukazatel a Parametr, a tím vytvořit popis Údajů. Podle umístění dimenzí do os současně systém vytváří základní strukturu Datové oblasti (viz obrázky níže).

Celkem existují tři základní typy Datových oblastí. V rámci jednoho Výkazu mohou být tyto typy kombinovány (to znamená, že se Výkaz může skládat z Datových oblastí všech níže popsaných typů):

- **statické:** zobrazuje se jako tabulka s přesně definovaným počtem řádků a sloupců daných konkretizací ukazatelů a parametrů. Údaj má pevnou pozici ve struktuře. Na osách x a y může být definováno 1 až n dimenzí, tj. Ukazatelů a Parametrů,

		Objem hypotečních úvěrů			Objem spotřebitelských úvěrů		
		Celkem	do splatnosti	po splatnosti	Celkem	do splatnosti	po splatnosti
A	B	1	2	3	4	5	6
Koruna česká	1						
Euro	2						
Dolar	3						
Libra	4						
Rubl	5						

Obrázek 8 - Příklad layoutu statické Datové oblasti

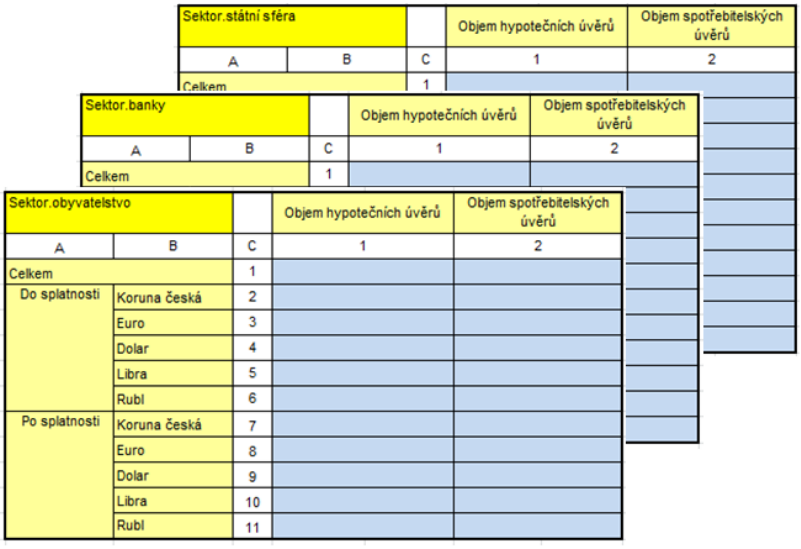
- **dynamické:** tabulky s proměnlivým počtem řádků nebo sloupců. Vykazující osoba zasílá Hodnoty údajů a k nim jednoznačně přiřazené hodnoty Parametrů v kombinacích, které se reálně vyskytly. Na osách x a y může být definováno 1 až n dimenzí, tj. Ukazatelů a Parametrů.

Dynamické Datové oblasti jsou typicky vytvářeny pro transakční data, např. údaje o jednotlivých úvěrech, cenných papírech, jednotlivé měny, země apod. Dynamická Datová oblast může obsahovat kromě dynamických údajů i údaje statické reprezentované součtovým řádkem,

			Objem hypotečních úvěrů			Objem spotřebitelských úvěrů		
			Celkem	do splatnosti	po splatnosti	Celkem	do splatnosti	po splatnosti
A	B	C	1	2	3	4	5	6
Součet		1						
Výčet zemí ...	Výčet měn	2						
.....	.....	...						
.....	.....	...						
....	.....	...						

Obrázek 9 - Příklad layoutu dynamické Datové oblasti

- **kartotékové:** statická nebo dynamická Datová oblast obsahující mimo osu x a y také osu z. Osa z je přidána jako třetí rozměr Datové oblasti a je určená Hierarchií číselníku, Doménou číselníku nebo Doménou datového typu definováním Konkretizovaného parametru nebo právě jedním Ukazatelem. Všechny osy mohou být popsány 1 až n dimenzemi, tj. Ukazateli a Parametry. Vykazující osoba zasílá Hodnoty údajů a v případě dynamických Údajů k nim jednoznačně přiřazené hodnoty vykazovaných Parametrů,



Obrázek 10 - Příklad layoutu kartotékové Datové oblasti

Systém umožňuje vytvářet klony instance objektu Datová oblast. Toto se použije v případě, že je potřeba jednu Datovou oblast sdílet ve více Výkazech. Není použit systém klasického sdílení, kde by existovala pouze jedna instance objektu Datová oblast a bylo na ni odkazováno ze dvou (a více) míst, ale systém „fiktivního sdílení“. Tento systém spočívá v tom, že v okamžiku potřeby sdílet Datovou oblast se vytvoří její klon (vznikne nová instance objektu Datová oblast). Klonován je jak objekt Datová oblast, tak i podřízené instance objektů (Údaj, Kontrola).

V rámci objektu Datová oblast je definována nepovinná asociační rekurzivní vazba s názvem „mateřská DO“. V případě, že nějaká Datová oblast vznikne jako klon jiné Datové oblasti,

pak nově vzniklá Datová oblast udržuje referenci na Datovou oblast, ze které vznikla (atribut u nově vzniklé instance objektu Datová oblast „mateřská DO“ bude v takovém případě obsahovat identifikátor instance Datové oblasti, ze které vznikl). Vazba je nepovinná, to znamená, že pokud Datová oblast nevznikla jako klon jiné Datové oblasti, je pak atribut „mateřská DO“ nastaven na hodnotu „NULL“ (atribut „mateřská DO“ vznikne až v reálném databázovém modelu; v objektovém modelu je tento atribut reprezentován výše popsanou vazbou, jedná se tak o odvozený atribut).

Platí následující pravidla:

- nelze schválit výkaz, který obsahuje Datovou oblast, která je klonem jiné Datové oblasti (mateřská Datová oblast), aniž by Výkaz, který obsahuje tuto mateřskou Datovou oblast, nebyl ve stavu Schválený,
- klonovat nějakou Datovou oblast je možno až ve stavu, kdy je Výkaz, který danou Datovou oblast obsahuje, ve stavu Schválený. Výjimku z tohoto pravidla tvoří situace, kdy uživatel u nějaké Datové oblasti nastaví hodnotu atributu „umožnit sdílení před schválením výkazu“ na hodnotu „ano“.

Existuje vazba mezi objektem Datová oblast a Kontrola. Jedná se o nepovinnou asociační vazbu, která je postavena tak, že jedna Datová oblast může obsahovat N Kontrol (tedy i žádnou) a jedna Kontrola může být navázána buď na žádnou, nebo na právě jednu Datovou oblast. Protože je vazba definována na předka všech typů kontrol (objekt Kontrola), je třeba zajistit, aby nebylo nutné definovat vazbu kontroly na Datovou oblast v případě, že se jedná o Mezivýkazové (MVK) a Mezisubjektové kontroly (MSK). V případě Jednovýkazových kontrol (JVK) a Kontrol časových řad (KČŘ) je vazba kontroly na Datovou oblast povinná.

Vztah objektu Datová oblast k objektu Blok výkazu je popsán výše.

Dále platí, že Datová oblast má vazbu na objekt Údaj, přičemž platí:

- Datová oblast může vzniknout, aniž by obsahovala alespoň jeden Údaj,
- Datová oblast může mít přiřazeno N (neomezeně) Údajů,
- pokud vzniká Údaj, musí být zařazen do právě jedné Datové oblasti,
- pokud dojde ke smazání Datové oblasti, pak musí dojít ke smazání všech souvisejících instancí objektu Údaj.

Vazba objektu Datová oblast na objekt Údaj je popsána v kapitole 3.18. Objekt Údaj.

3.4.1 Atributy objektu Datová oblast

Objekt Datová oblast obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů).

Nad rámec standardních atributů jsou u objektu Datová oblast definovány následující atributy:

- **typ datové oblasti (number; číselníková položka):** hodnota atributu je tvořena identifikátorem číselníkové položky. Číselník je statický (není definovatelný uživatelem) a je tvořen těmito hodnotami:
    - statická,
    - dynamická,
    - kartotéková.

Hodnota atributu je stanovována uživatelem na začátku projektování Datové oblasti podle charakteru zadaných konkretizací dimenzionálních Parametrů. Pro každý typ datové oblasti jsou nastavena pravidla projektování podle typu zvolené Datové oblasti,

- **možnost sdílení před schválením výkazu (boolean):** určení, zda je možné konkrétní Datovou oblast sdílet před schválením výkazu, hodnota „ano“/“ne“.

3.5 Objekt Číselník

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Účelem objektu Číselník je možnost zastřešovat strukturované seznamy položek, případně hierarchie těchto položek.

Objekt Číselník (a s ním související objekty) se týká pouze problematiky metapopisu, tj. slouží k zachycení metadat, která definují Výkazy. V textu ostatních dokumentů zadání SDAT jsou dále obecným pojmem „číselník“ označeny obory hodnot atributů objektů různých komponent systému. Tyto číselníky jsou také nazývány jako systémové číselníky a nejsou předmětem této kapitoly.

Obsahem každého Číselníku je smysluplná konečná množina Položek číselníku potřebných pro definování Údajů požadovaných ve výkazech.

Číselník se skládá z Položek číselníku. Mezi těmito položkami nejsou definovány žádné vazby nadřízenosti a podřízenosti položek. Pokud je třeba takové vazby definovat, pak je nutné Položky číselníku seskupit do hierarchií, které umožní tyto vazby definovat. V rámci hierarchie číselníku pak vznikají součtové položky číselníkových položek (uzlové položky). V takovém případě jsou vytvářeny instance objektu Hierarchie číselníku (viz kapitola 3.7 Objekt Hierarchie číselníku)

Nad Položkami číselníku nebo Položkami hierarchie může být vytvořena jedna nebo více Domén číselníku. Doména číselníku je tak vlastně podmnožina Položek číselníku/Položek hierarchie, která dohromady tvoří nějaký smysluplný celek. Pokud je Doména číselníku postavena nad Položkami číselníku, mluvíme o doméně vzniklé z číselníku, pokud je postavena nad Položkami hierarchie, mluvíme o doméně vzniklé z hierarchie (viz kapitola 3.9.1 Atributy objektu Doména číselníku).

Číselník obsahuje veškeré položky, tj. elementární i součtové, vzniklé ze zařazení elementárních položek do hierarchie, na stejné úrovni. Každá součtová položka je v přehledu Položek číselníku viditelně označena. Systém umožní oddělené zobrazení elementárních a součtových položek, u součtových položek musí být automaticky možnost zobrazení přehledu elementárních položek do součtové položky zahrnutých (viz kapitola 3.6.1 Atributy objektu Položka číselníku a 3.8.1 Atributy objektu Položka hierarchie).

Číselníky můžeme rozlišit následovně:

a) z pohledu původu Číselníku:
   - číselníky mezinárodní (např. číselník zemí a číselník měn, ESA2010, odvětví NACE),
   - číselníky národní (číselník právních forem, číselník okresů, číselníky vycházející z norem např. cenné papíry),

- číselníky ČNB (specifické číselníky vytvářené v ČNB),
- kombinované (číselníky vycházejí z mezinárodních i národních požadavků).

b) z pohledu atributu správy Číselníku
- globální číselníky, tj. číselníky průřezové, které jsou využívány zpravidla více výkazy,
- lokální číselníky, tj. číselníky, které jsou využívány jen pro jeden nebo několik výkazů (o přiřazení atributu rozhoduje dle předchozí analýzy uživatel (správce)).

Každý Číselník při svém založení obsahuje množinu Položek číselníku na jedné úrovni (tj. na začátku Číselník obsahuje jen elementární položky). Postupně mohou být přidávány další elementární položky a součtové (uzlové) položky. Položka se stává součtovou (uzlovou) po zařazení do hierarchie a definování jejího elementárního obsahu.

Objekt Číselník lze vytvořit různými způsoby:

- ručním typováním Položek číselníku s povinnými atributy,
- natažením Číselníku resp. Položek číselníku ze souboru s přesně definovaným rozhraním (určeno pro takové číselníky, které jsou přebírány z externích zdrojů např. ČSÚ, ISO normy, DPM).

V případě změny Číselníku musí systém identifikovat všechny dopady na objekty, které jsou danou změnou zasaženy (např. ukončení Položky číselníku má dopad na Hierarchie číselníku, Domény číselníku, Datovou oblast, Kontroly apod.; podrobněji o závislostech objektů pojednává kapitola 2.3 Vazby mezi jednotlivými objekty).

Instance objektu Číselník vzniká bez vazby na jakýkoli jiný objekt nebo instanci objektu.

Objekt Číselník má vazbu na objekt Položka číselníku:

- Číselník může vzniknout, aniž by obsahoval alespoň jednu Položku číselníku,
- Číselník může obsahovat N (neomezeně) Položek číselníku,
- pokud vzniká Položka číselníku, pak musí být přiřazena k právě jednomu Číselníku,
- konkrétní Položka číselníku je přiřazena právě jednomu Číselníku (neexistuje sdílení číselníkových položek, pokud je třeba nějakou Položku číselníku mít v různých Číselnících, je tato Položka číselníku vytvořena znovu, tj. je tak v systému redundantně),
- v případě potřeby změnit zařazení Položky číselníku k Číselníku, dojde k přepsání informace o tom, do jakého Číselníku byla Položka číselníku zařazena (Položka číselníku „POL1“ má původně vazbu na Číselník„CIS1“ a po změně má vazbu na „CIS2“, přičemž dojde ke ztrátě informace, že v minulosti měla Položka číselníku „POL1“ vazbu na Číselník „CIS1“),
- pokud dojde ke smazání instance objektu Číselník, dojde ke smazání všech souvisejících instancí objektu Položka číselníku,
- pokud dojde ke smazání Položek číselníku, nedojde ke smazání instance Číselník,
- pokud mají být v rámci Číselníku definovány Hierarchie číselníku, musí nejdříve existovat instance objektu Položka číselníku, aby tyto Hierarchie mohly vzniknout (viz kapitoly 3.7 Objekt Hierarchie číselníku a 3.8 Objekt Položka hierarchie).

Objekt Číselník má vazbu na objekt Hierarchie číselníku:

- Číselník může vzniknout a trvale existovat, aniž by obsahoval alespoň jednu instanci objektu Hierarchie číselníku,
- Číselník může obsahovat N (neomezeně) instancí Hierarchie číselníku,

- pokud vzniká instance Hierarchie číselníku, pak musí být přiřazena právě jednomu Číselníku a po celou dobu životnosti je instance objektu Hierarchie číselníku přiřazena právě jednomu Číselníku,
- Hierarchie číselníku nelze přesouvat mezi Číselníky (plyne z logiky věci),
- pokud dojde ke smazání instance objektu Číselník, dojde ke smazání všech souvisejících instancí objektu Hierarchie číselníku (a následně i všech souvisejících instancí objektu Položka Hierarchie).

Objekt Číselník má vazbu na objekt Doména číselníku:

- účelem této vazby je definovat, nad jakým Číselníkem je Doména číselníku (resp. lépe řečeno doména číselníkových položek nebo doména položek hierarchie) vybudována. Pokud bude tato informace známa, bude umožněno omezit při tvorbě Domény číselníku nabídku Položek číselníku pouze na ty položky, které patří do jednoho Číselníku a splnit tak požadavek, že v rámci jedné Domény číselníku nesmí být použity Položky číselníku z více různých Číselníků,
- Číselník může vzniknout a trvale existovat, aniž by obsahoval alespoň jednu instanci objektu Doména číselníku,
- jeden Číselník může mít N (neomezeně) Domén číselníku,
- jedna Doména číselníku je vytvořena v rámci právě jednoho Číselníku. Doména číselníku nemůže vzniknout, pokud neexistuje Číselník,
- v případě, že dojde ke smazání Číselníku, dojde ke smazání všech Domén číselníku, které jsou v rámci mazaného Číselníku vytvořeny.

Objekt Číselník má vazbu na objekt Převodník, ale s ohledem na specifičnost vazby objektů Číselník/Převodník jsou tyto vazby popsány u objektu Převodník (viz kapitola 3.10 Objekt Převodník).

3.5.1 Atributy objektu Číselník

Objekt Číselník obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů).

Nad rámec standardních atributů jsou u objektu Číselník definovány tyto atributy:

- **atribut pro správu objektu:** umožňuje definovat, zda je Číselník (resp. jeho obsah, tj. Položky číselníku a Hierarchie číselníku z nich odvozené) spravován na globální úrovni (uživatel v roli „správce číselníků“) nebo je ve správě určených (lokálních) uživatelů. Možné hodnoty v tomto atributu jsou „globální“ a „lokální“,
- **UM pro správu:** umožňuje definovat uživatelské místo, které opravňuje uživatele, kteří jsou na něj přiřazeni s časovou platností k úpravě obsahu Číselníku. Tento atribut má smysl jen v případě, že je hodnota atributu „atribut pro správu objektu“ rovna hodnotě „lokální“,
- **Atribut pro proces schvalování naplnění lokálního číselníku:** umožňuje definovat, v jaké fázi procesu schvalování se nachází. Prvotní naplnění lokálního číselníku je schvalováno správcem číselníků a atribut objektu prochází stavy „Čeká na potvzení“, „Potvrzený“ nebo „Nepotvrzený“.

Proces vzniku lokálního a globálního Číselníku je popsán v kapitole 5.4 Proces tvorby objektů popisujících údaje.

3.6 Objekt Položka číselníku

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Účelem objektu Položka číselníku je evidence položek seznamů, které se v aplikaci používají opakovaně a u nichž je vyžadováno zcela přesné zadání ze strany uživatele. Typickým příkladem je například číselník „Měny“ a jeho položka pak je například „CZK“, což je zkratka české národní měny, která vychází z mezinárodní normy ISO 4217.

Položka číselníku nemá verze ani varianty, nemůže existovat bez Číselníku. Položky číselníku mohou být:

- **elementární:** položka, která nemá další rozpad (podřízené položky),
- **součtové (uzlové)**: obsah položky je součtem elementárních položek. Součtovou se položka stává při zařazení do hierarchie, kde je definován její elementární rozpad.

Každá součtová položka má atribut „suma“ nastavený na „ano“ a znak ∑ viditelný na úrovni seznamu Položek číselníku, zároveň je umožněno oddělené zobrazení elementárních a součtových položek. Atribut suma systém automaticky nastaví Položce číselníku po jejím zařazení do Hierarchie číselníku. Z důvodu přehlednosti jsou stanovena metodická pravidla mnemotechnického přidělování kódu součtových položek. Uživateli je umožněno zobrazit obsah součtové položky (rozklad do elementárních položek) ze základního přehledového formuláře, který obsahuje přehled Položek číselníku.

Vztah objektu Položka číselníku k objektu Číselník je popsán výše.

Vazba objektu Položka číselníku na objekt Položka hierarchie má následující pravidla:

- pokud v rámci Číselníku mají být definovány nějaké vazby nadřízenosti a podřízenosti mezi Položkami číselníku, pak tyto hierarchie lze postavit, pokud jsou nejdříve nadefinovány příslušné Položky číselníku (například v Číselníku zemí chceme seskupit země do vyššího regionálního uspořádání typu Evropa = ČR, SR, Německo a další, Afrika = Egypt, Súdán, Tunisko a další), Položkou číselníku se tak musí nejdříve stát „Evropa“ a „ČR“, aby následně bylo možno nadefinovat hierarchickou vazbu „ČR je součástí vyššího celku Evropa“),
- jedna Položka číselníku se může stát více Položkami hierarchie pouze tehdy, pokud je v rámci Číselníku vytvořeno více Hierarchií číselníku. V rámci jedné konkrétní Hierarchie číselníku se smí Položka číselníku objevit maximálně jednou,
- může existovat Položka číselníku, která zároveň není zařazena do žádné Hierarchie číselníku,
- pokud by došlo ke smazání instance objektu Položka číselníku, musí být smazány i všechny související instance objektu Položka hierarchie,
- pokud by došlo k přesunutí Položky číselníku z jednoho Číselníku do jiného, pak dojde ke smazání všech souvisejících Položek hierarchie (musí být dodrženo pravidlo, že jedna hierarchie je vždy vybudována nad Položkami číselníku z právě jednoho Číselníku).

3.6.1 Atributy objektu Položka číselníku

Objekt Položka číselníku obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů) s výjimkou následujících:

- Garant.

Nad rámec standardních atributů nejsou u objektu Položka číselníku definovány žádné další atributy.

3.6.2 Dynamické atributy objektu Položka číselníku

Kromě standardních atributů, které jsou definovány v rámci objektu Položka číselníku, systém umožňuje definovat tzv. dynamické atributy. Dynamické atributy jsou další (dodatečné) atributy, které je možné definovat ke každé Položce číselníku/hierarchie.

Definice těchto atributů je v modelu umožněna přes objekt Dynamický atribut číselníku. Tento objekt je napojen kompoziční vazbou na objekt Číselník v kardinalitě 1:N. To znamená, že každý jeden Číselník může mít definováno N (tedy i žádný) dynamických atributů a jeden dynamický atribut je vždy připojen vždy k právě jednomu Číselníku (není tak umožněno sdílení dynamických atributů mezi více Číselníky). V případě, že dojde ke smazání instance objektu Číselník, jsou smazány všechny existující související instance objektu Dynamický atribut číselníku.

V rámci definice Dynamického atributu číselníku je třeba definovat:

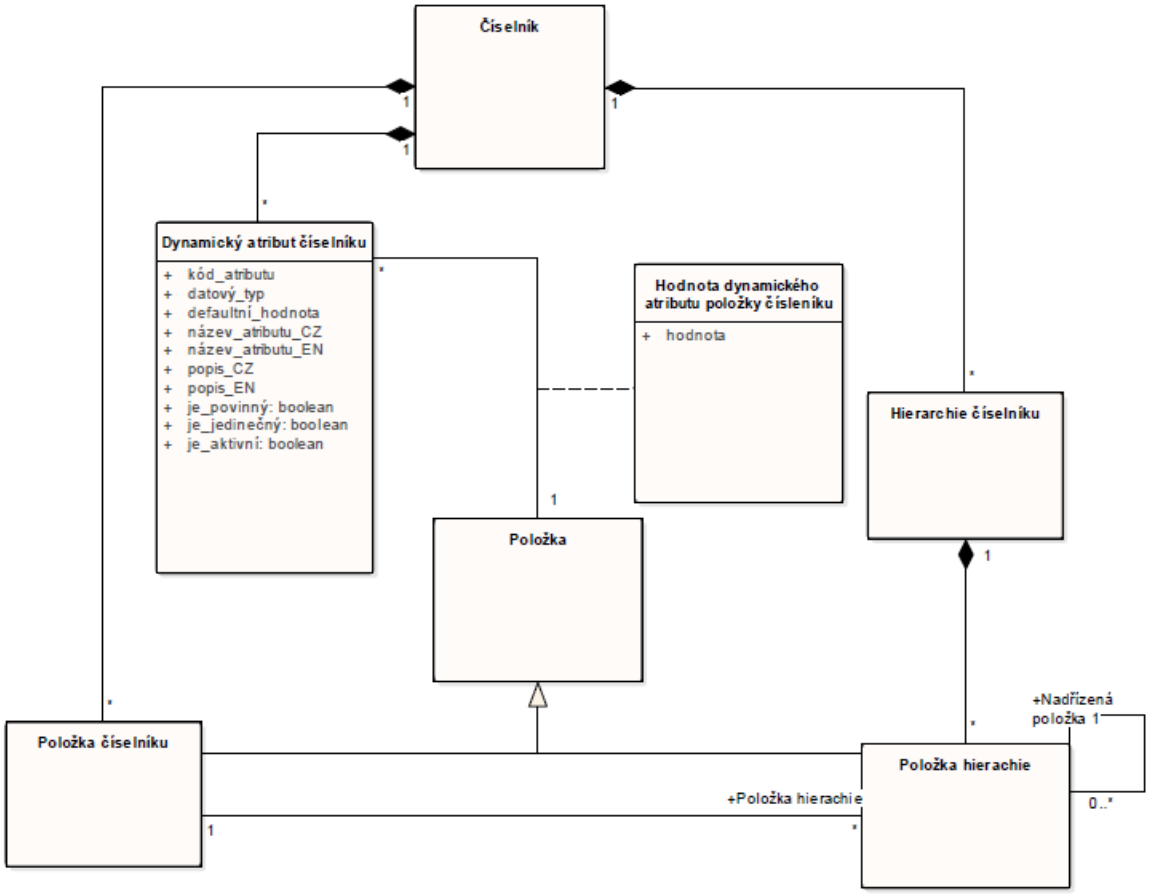
- Datový typ: určuje, jaké hodnoty jsou povoleny pro daný atribut. Jsou k dispozici pouze tyto základní datové typy:
  - NUMBER
  - STRING
  - DATE,
- defaultní hodnota: umožní definovat, jaká hodnota se bude v daném atributu defaultně nastavovat při vytváření nové Položky číselníku. Tento atribut musí obsahovat nějakou validní hodnotu v případě, že je vytvářen nový dynamický atribut, který je označen jako povinný a již existují nějaké Položky číselníku. V takovém případě bude tato hodnota použita u všech již existujících Položek číselníku,
- je povinný (default: ne): pokud je definováno „ano“, není možno vytvořit Položku číselníku bez definice validní hodnoty tohoto atributu,
- je jedinečný (default: ne): pokud bude definováno „ano“, pak to znamená, že nemohou existovat dvě číselníkové položky v rámci jednoho Číselníku, které by v daném atributu obsahovaly stejnou hodnotu,
- je aktivní (default: ano): pokud bude definováno „ano“, je daný atribut nabízen při tvorbě nové číselníkové položky, v opačném případě nikoli.

Výše uvedený text popisuje definici dynamických atributů. Objekt Hodnota dynamického atributu pak umožňuje uložit uživatelem zapsanou hodnotu daného dynamického atributu ke konkrétní Položce číselníku. Objekt Hodnota dynamického atributu je definován v rámci asociační vazby mezi objekty Dynamický atribut číselníku a Položka (abstraktní objekt, předek Položky číselníku a Položky hierarchie) s kardinalitou 1:N, což znamená, že jeden dynamický atribut se vždy vztahuje k právě jedné instanci objektu Položka a jedna instance

objektu Položka smí mít definováno neomezeně dynamických atributů (neomezeně zde znamená až do výše počtu dynamických atributů definovaných pro Číselník, kde se Položka číselníku vyskytuje).

Pokud jsou pro nějaký Číselník nadefinovány dynamické atributy, pak platí pro všechny jeho Položky číselníku. Dále platí, že každá Položka číselníku může mít pro konkrétní dynamický atribut maximálně jednu hodnotu (buď žádnou, nebo jednu; není tedy možné k jedné Položce číselníku a jednomu dynamickému atributu uchovávat více hodnot, například s různou časovou platností).

3.6.2.1 *Objektový model pro definici dynamických položek číselníku*



Obrázek 11 - Část modelu metapopisu zachytávající možnost definovat dynamické atributy Položek číselníků a ukládat jejich hodnoty

3.7 Objekt Hierarchie číselníku

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Účelem objektu Hierarchie číselníku je umožnit definovat vazby (nadřízenost a podřízenost) jednotlivých instancí objektu Položka číselníku. Tyto vazby jsou vytvořeny prostřednictvím objektu Položka hierarchie.

V rámci objektu Hierarchie číselníku definujeme tyto pojmy:

- **kořenová položka:** vrcholová Položka hierarchie, která má další rozpad v rámci Hierarchie číselníku. Kořenová položka je vždy označena znakem (∑) a je v rámci Hierarchie číselníku právě jedna,
- **kořen hierarchie:** obsahuje kořenovou položku včetně celé stromové struktury Hierarchie číselníku až do úrovně elementárních položek a je v rámci Hierarchie číselníku právě jeden,
- **uzlová položka:** nadřízená Položka hierarchie, která má další rozpad uvnitř Hierarchie číselníku; uzlová položka je vždy označena znakem (∑),

- **uzel hierarchie:** obsahuje uzlovou položku včetně její celé stromové struktury jí podřízených Položek hierarchie až do úrovně elementárních položek. Uzel hierarchie může obsahovat N (neomezeně) uzlů hierarchie, jež jsou mu podřízené,
- **elementární položka:** podřízená Položka hierarchie, tj. Položka hierarchie, která nemá další rozpad a tvoří poslední úroveň Hierarchie číselníku,
- **úroveň hierarchie:** úroveň zanoření Položky hierarchie v rámci Hierarchie číselníku.

Hierarchie číselníků lze rozdělit na:

- **globální:** hierarchie tvořené nad globálním Číselníkem,
- **lokální:** hierarchie tvořené nad lokálním Číselníkem.

To, zda je Hierarchie číselníku lokální nebo globální, je informace, která je zděděna od související instance objektu Číselník, není ji tedy třeba znovu definovat v rámci objektu Hierarchie číselníku. Pro editaci obsahu Hierarchie číselníku z hlediska lokálnosti/globálnosti dané Hierarchie číselníku tak platí stejná pravidla jako pro související instanci objektu Číselník.

Obecné vlastnosti objektu Hierarchie číselníku jsou:

- Hierarchie číselníku může vzniknout, aniž by obsahovala alespoň jednu Položku hierarchie,
- Hierarchie číselníku, která obsahuje méně než tři Položky hierarchie, nemá metodický význam,
- každá Hierarchie číselníku obsahuje 2 až N úrovní hierarchie,
- Hierarchie číselníku může obsahovat N (neomezeně) Položek hierarchie v N (neomezeně) úrovních hierarchie,
- každá Hierarchie číselníku může obsahovat N (neomezeně) uzlů hierarchie, přičemž každý uzel hierarchie začíná právě jednou uzlovou položkou,
- každá Položka hierarchie musí být přiřazena k právě jedné Hierarchii číselníku,
- každá Položka hierarchie musí být přiřazena k právě jedné Položce číselníku,
- každá Položka hierarchie je v dané Hierarchii číselníku právě jednou,
- k jedné Položce číselníku se může vázat N (neomezeně) Položek hierarchie,
- pro jeden Číselník může být vytvořen neomezený počet Hierarchií číselníku, ale jedna Hierarchie číselníku je vytvářena zásadně jen nad jedním Číselníkem,
- pokud dojde ke smazání instance objektu Hierarchie číselníku, dojde ke smazání všech souvisejících instancí objektu Položka hierarchie,
- systém umožňuje jak stromové mazání uzlové položky, tak smazání uzlové položky se zachováním jejího obsahu:
    - v případě, že je uživatelem zvolena metoda stromového mazání, pak platí, že se smazáním uzlové položky jsou smazány všechny její podřízené položky ve všech úrovních podřízenosti, bez ohledu na to, zda jsou elementární nebo uzlové,
    - v případě, že je uživatelem zvolena metoda mazání uzlové položky se zachováním jejího obsahu, pak systém nejdříve pro položky na úrovni N-1 (bráno od mazané uzlové položky) změní odkaz na nadřízenou položku, a to tak, že jako nového rodiče nastaví položku, která je přímo nadřízená mazané položce. Tím je dosaženo toho, že je obsah mazané uzlové položky přesunut pod jiného rodiče a smazání uzlové položky nevyvolá smazání jejího obsahu,
- pokud je smazána Položka hierarchie, nemá to vliv na Položku číselníku,

- každé uzlové Položce hierarchie odpovídá právě jedna součtová Položka číselníku. V Číselníku je pak taková položka označena znakem (∑) a je možné si přímo z Číselníku zobrazit její elementární rozpad,
- v rámci jednoho Číselníku smí být jen jedna Hierarchie číselníku nesynchronizovaná, tedy atribut Hierarchie číselníku „Provedení synchronizace“ je nastaven na „Ne“ (viz kapitola 3.7.2 Proces tvorby Hierarchií číselníku). To neznamená, že v rámci jednoho Číselníku nemůže být více Hierarchií číselníku ve stavu Projektovaný. Systém nepovolí vytvoření nebo úpravu (vytvoření verze) další Hierarchie číselníku, pokud všechny Hierarchie číselníku vytvořené nad daným Číselníkem nejsou synchronizovány. Důvodem je dodržení konzistence obsahu všech souvisejících uzlů hierarchie ve všech Hierarchiích číselníku,
- systém indikuje existenci uzlové položky, pokud již existuje uzlová položka se stejným elementárním rozkladem pod jiným kódem (omezení duplicit). Uživatel povolí nebo nepovolí vznik takové duplicitní uzlové položky,
- editace atributů objektu Hierarchie číselníku (název, kód, definice a popis) se řídí pravidly popsanými v kapitole 2.3 Vazby mezi jednotlivými objekty,
- vytvoření Hierarchie číselníku je dvoustupňový proces, který obsahuje aktualizaci a synchronizace (viz kapitola 3.7.2 Proces tvorby Hierarchií číselníku).

3.7.1 Atributy objektu Hierarchie číselníku

Objekt Hierarchie číselníku obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů). Nad rámec standardních atributů jsou u objektu Hierarchie číselníku definovány atributy:

- **atribut provedení synchronizace:** atribut nabývá hodnot „ano/ne“. Při založení hierarchie je tento atribut nastaven defaultně na hodnotu „ne“. Tento atribut je automaticky řízen systémem, uživatel nemůže přepsat hodnotu atributu. Může ji však ovlivnit akcí provedení synchronizace,
- **atribut pro správu objektu:** hodnota atributu je zděděna od atributu definovaného na instanci Číselník.

3.7.2 Proces tvorby Hierarchií číselníku

Pro práci s objektem Hierarchie číselníku platí:

- uživatel založí prázdný objekt Hierarchie číselníku a vyplní mu všechny potřebné atributy,
- při vkládání dalších Položek hierarchie uživatel volí, na kterou úroveň je Položka hierarchie vkládána za pomocí nástroje systému,
- systém umožňuje při zobrazení Hierarchie číselníku rozlišit jednotlivé úrovně hierarchie,
- systém umožňuje při zobrazení Hierarchie číselníku řadit Položky hierarchie v rámci jedné úrovně jednoho uzlu hierarchie abecedně, případně v pořadí určené uživatelem,
- systém umožňuje uživateli označit při zobrazení Hierarchie číselníku jednotlivé uzly hierarchie, elementární položky a skupiny elementárních položek na jedné úrovni jednoho uzlu hierarchie a přesouvat je v rámci dané Hierarchie číselníku,
- uživatel může vytvořit instanci objektu Hierarchie číselníku jako replikaci kořene hierarchie nebo uzlu hierarchie jiné Hierarchie číselníku,

- uživatel může vytvářet Hierarchii číselníku po částech tak, že jsou postupně ukládány části Hierarchie číselníku,
- uživatel může importovat Hierarchii číselníku z jiného zdroje a systém zaručí, že jsou při tomto importu dodržena všechna pravidla tvorby Hierarchie číselníku. Pro import hierarchie platí následující:
    - Import Hierarchie číselníku je možný pouze v případě, kdy v době importu Hierarchie číselníku v systému SDAT neexistuje žádná Hierarchie daného číselníku.
    - Zároveň importem lze importovat pouze jednu Hierarchii číselníku.
    - Pokud importovaná Hierarchie číselníku obsahuje Položky číselníku, které v době importu neexistují v Číselníku, pak je na ně pohlíženo jako na nové, resp. jako na položky, které budou přidány, viz funkční požadavek (CIS_16_0 část 1) i a část 2) i).
    - Importovaná Hierarchie číselníku musí projít procesem aktualizace a synchronizace.

Po vytvoření nebo modifikaci Hierarchie číselníku je nutné provést aktualizaci a synchronizaci instancí Hierarchie číselníku.

3.7.2.1 Aktualizace instance Hierarchie číselníku

V rámci aktualizace dochází k zajištění konzistence mezi objekty, které přímo souvisí s vytvářenou Hierarchií číselníku. Aktualizace je provedena v okamžiku uložení instance Hierarchie číselníku. V rámci aktualizace systém provádí následující akce:

- eviduje informace o obsahu uzlů vytvářené nebo modifikované Hierarchie číselníku a jejich elementárních položkách,
- eviduje informaci o vzniku nové uzlové položky z elementární položky nebo naopak a propaguje tuto informaci do Číselníku:
    - v případě vzniku nové uzlové položky z elementární položky přidá znak (∑) k Položce číselníku (tj. vytvoří součtovou Položku číselníku),
    - v případě změny uzlové položky na elementární položku plošně ve všech instancích Hierarchie číselníku odebere znak (∑) Položce číselníku (tj. vytvoří elementární Položku číselníku),
- zobrazí uživateli diagnostiku dopadu, tj. seznam objektů, na něž má (přímo nebo nepřímo, podle závislosti objektů podle kapitoly 2.3 Vazby mezi jednotlivými objekty) dopad vytvářená Hierarchie číselníku včetně popisu tohoto dopadu. Diagnostiku je možné nejen zobrazit, ale i uložit, případně vytisknout,
- umožní uživateli potvrdit, zda souhlasí nebo nesouhlasí s avizovaným dopadem,
    - v případě nesouhlasu s avizovanými změnami umožní uživateli znovu provést standardní úpravy dané Hierarchie číselníku, tedy umožní i neuložit změny a navrátit se k původní hierarchii před aktualizací,
    - v případě souhlasu uživatele s avizovaným dopadem je daná Hierarchie číselníku aktualizována.

3.7.2.2 Synchronizace instance Hierarchie číselníku

V rámci synchronizace dochází k zajištění souladu všech instancí objektu Hierarchie číselníku, jež jsou vytvořeny nad jednou instancí objektu Číselník tak, aby elementární rozklad společné uzlové nebo kořenové položky se shodoval ve všech Hierarchiích číselníku vytvořených nad jedním Číselníkem:

- synchronizace je prováděna na pokyn uživatele v momentě, kdy je správně dokončena aktualizace (viz kapitola 3.7.2.1 Aktualizace instance Hierarchie číselníku). V rámci synchronizace systém provádí následující akce:
  - porovná obsah všech souvisejících uzlů hierarchií pro jeden Číselník,
  - zobrazí uživateli diagnostiku změn v souvisejících uzlech hierarchií, které je nutné provést, aby byly stávající uzly hierarchie uvedeny do souladu s nově vytvořenými uzly hierarchie (včetně přehledu verzí/variant Hierarchií číselníků které v důsledku těchto změn systém vytvoří nebo se modifikují). Diagnostiku je možné nejen zobrazit, ale i uložit, případně vytisknout,
  - zobrazí uživateli diagnostiku dopadu na ostatní objekty, které budou změnou dané Hierarchie číselníku dotčeny (Domény číselníku, Datové oblasti, Ukazatele, Výkazy, kontroly), včetně informace, zda budou vytvořeny nové verze či varianty dotčených instancí objektů (Doména číselníku, Datová oblast, Ukazatel, Výkaz, kontroly) nebo zda budou modifikovány. Diagnostiku je možné nejen zobrazit, ale i uložit, případně vytisknout,
  - v případě, že byla při vytvoření nebo modifikaci Hierarchie číselníku provedena akce, která vede ke konfliktu (např. záměna uzlů, duplicita položek (viz příklady v kapitole 8.2 Příloha 2 - Příklady pro pravidla aktualizace a synchronizace Hierarchií číselníku), systém uživateli ohlásí konflikt, který musí uživatel vyřešit ručně další úpravou vytvářené nebo modifikované Hierarchie číselníku,
  - umožní uživateli potvrdit, zda souhlasí, nebo nesouhlasí s avizovaným dopadem:
    - v případě přijetí změn uloží nově vytvořené verze/varianty instancí objektu Hierarchie číselníku a také nové verze/varianty objektů, na něž má (přímo nebo nepřímo) vytvoření těchto verzí/variant instancí objektu Hierarchie číselníku dopad, a modifikuje obsah,
    - v případě nepřijetí změn, uživatel musí upravit zdrojovou Hierarchii číselníku, tak, aby byla věcně správně a odpovídala pravidlům pro vytváření objektu Hierarchie číselníku,
  - když je synchronizace správně dokončena, je dané Hierarchii číselníku nastaven atribut provedení synchronizace na „ano“,
  - pokud není synchronizace dokončena, zůstává atribut provedení synchronizace nastaven na defaultní hodnotu „ne“,
- pokud je docíleno stavu, že v rámci jednoho Číselníku mají všechny Hierarchie číselníku nastaven atribut provedení synchronizace na „ano“, je možné vytvářet nebo modifikovat další Hierarchii číselníku, která je vytvořena nad stejnou instancí objektu Číselník,
- pokud nad jednou instancí objektu Číselník existuje nějaká instance objektu Hierarchie číselníku, která má atribut provedení synchronizace nastaven na „ne“, informuje systém uživatele o nutnosti její synchronizace. Do té doby systém neumožní uživateli vytvořit nebo upravit další Hierarchii číselníku, vytvořenou nad stejnou instancí objektu Číselník. Součástí informace je kód a název této instance objektu Hierarchie číselníku.

Při aktualizaci a synchronizaci Hierarchií číselníku je kontrolováno dodržení následujících základních pravidel (příklady k níže uvedeným pravidlům jsou uvedeny v kapitole 8.2 Příloha 2 - Příklady pro pravidla aktualizace a synchronizace Hierarchií číselníku):

1. Při vložení elementární položky do Hierarchie číselníku dochází ke změně množiny elementárních položek všech nadřízených uzlových položek vložené elementární položky v dané Hierarchii číselníku. Tato změna se musí promítnout do všech Hierarchií číselníku, kde se tyto uzlové položky vyskytují, přičemž mohou vznikat konflikty (viz příklady v příloze 8.2.9 Konflikt typu Duplicita v nezávislých uzlech).
2. Při smazání elementární položky z Hierarchie číselníku dochází k jejímu smazání z množiny elementárních položek všech nadřízených uzlových položek v této Hierarchii číselníku, které v dané Hierarchii číselníku zůstaly. Tato změna se musí promítnout do všech souvisejících Hierarchií číselníku, kde se dané uzlové položky vyskytují.
3. Při smazání uzlové položky (bez smazání jejího elementárního obsahu) z aktualizované Hierarchie číselníku nedochází ke změně množiny elementárních položek této uzlové položky a do ostatních hierarchií se smazání uzlové položky nepromítá. Toto smazání je možno ignorovat.
4. Při smazání uzlové položky včetně jejího elementárního obsahu z aktualizované Hierarchie číselníku nedochází ke změně množiny elementárních položek této uzlové položky a do ostatních souvisejících Hierarchií číselníku se promítá změna elementárního obsahu v souvisejících uzlech hierarchie. Při synchronizaci dojde ke sladění obsahu souvisejících uzlových položek.
5. Vložení uzlové položky v aktualizované hierarchii pod jinou uzlovou položku se nepromítá do ostatních Hierarchií číselníku. Samotné toto vložení je možno ignorovat, musí však být provedena kontrola, zda neexistuje nějaká související Hierarchie číselníku, kde je vztah obou položek opačný (tzv. záměna uzlů). Pokud takový případ existuje, synchronizace se nepovolí. Při vložení uzlové položky do aktualizované Hierarchie číselníku však může dojít ke změně množiny elementárních položek této uzlové položky. Tento rozdíl se musí zjistit (dále se uplatňují pravidla 1 a 2).
6. Ke změně
   a) uzlové položky na elementární může dojít pouze v rámci aktualizace smazáním všech jejích podřízených Položek hierarchie, tzn. nemůže vzniknout nová elementární položka z uzlové položky v synchronizované hierarchii (souvisí s pravidlem 7).
   Pokud se položka, která se stala v rámci aktualizace hierarchie elementární z uzlové položky, také vyskytuje v některé související Hierarchii číselníku, tak v této Hierarchii číselníku zůstane. Toto může způsobit sekundární vložení této položky do některé další uzlové položky,

   b) elementární položky na uzlovou může dojít vložením Položek hierarchie pod jinou Položku hierarchie (tzn. vytvoření další úrovně hierarchie pod již existující Položkou hierarchie). Toto vložení ovlivní elementární úroveň souvisejících Hierarchií číselníku v souvisejících uzlech hierarchie. Pokud se z elementární položky stane uzlová položka, je toto indikováno v Číselníku znakem (∑).

7. V rámci synchronizace se smaže uzlová položka z Hierarchie číselníku pouze a právě v případě, pokud jsou smazány všechny její podřízené položky i položka sama a nemůže se stát elementární položkou podle pravidel 6.

Pokud se smaže uzlová položka z Hierarchie číselníku, ukončí se platnost odpovídající automatické Domény číselníku (viz kapitola 3.9 Objekt Doména číselníku), pokud je generována uživatelem. Pokud je však tato Doména číselníku použita (např. v konkretizaci Datové oblasti), nelze platnost ukončit. V daném případě se pouze smažou položky z dané Domény číselníku, smaže se příslušnost této Domény číselníku k Hierarchii číselníku a Uživatel dostane informaci o změně struktury příslušné Datové oblasti. Uživatel pak musí problém ve struktuře Datové oblasti vyřešit např. vytvořením a použitím jiné Domény číselníku.

3.7.2.3 Konflikty při Synchronizaci instance Hierarchie číselníku

Konfliktní situace mohou vznikat např. v případě přidání Položky instance Hierarchie číselníku:

a) pokud by jedna elementární položka byla obsažena dvakrát a vícekrát v jedné Hierarchii číselníku,
b) pokud by elementární rozpad stejné součtové položky byl různý ve více Hierarchiích číselníku.

Konflikty mohou vznikat také mezi:

- **závislými uzly**: stejná položka se vyskytuje duplicitně ve dvou závislých uzlech. Součtové položky, ve kterých se duplicitní položka vyskytuje, jsou ve vzájemném vztahu nadřízenosti nebo podřízenosti (viz kapitola 8.2.10 Konflikt typu Duplicita v závislých uzlech),
- **nezávislými uzly**: stejná položka se vyskytuje duplicitně ve dvou nezávislých uzlech. Součtové položky, kde se duplicitní položka vyskytuje, nejsou ve vzájemném vztahu nadřízenosti a podřízenosti (viz kapitola 8.2.9 Konflikt typu Duplicita v nezávislých uzlech).

3.8 Objekt Položka hierarchie

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Pokud mají být v rámci Číselníku definovány Hierarchie číselníku, musí nejdříve existovat instance objektu Položka číselníku, aby mohla vzniknout instance třídy Položka hierarchie. Hierarchii číselníku (resp. jejích položek) má tak smysl budovat pouze tehdy, pokud objekt Číselník obsahuje nějaké instance objektu Položka číselníku.

Účelem objektu Položka hierarchie je nadefinovat vazby mezi jednotlivými instancemi objektu Položka číselníku. Zatímco objekt Položka číselníku eviduje prostý výčet všech číselníkových položek (mezi položkami nejsou žádné vazby, v tomto objektu je pak „Evropa“ na stejné úrovni jako „ČR“), objekt Položka hierarchie už mezi těmito položkami vytváří vztah závislosti typu „nadřízená/podřízená položka“. Tento vztah určuje, že jedna instance objektu Položka hierarchie je nadřízená jiné instanci objektu Položka hierarchie, tedy že Položka hierarchie „ČR“ má definovánu nadřízenou Položku hierarchie „Evropa“.

Objekt Položka hierarchie se váže k objektu Položka číselníku následujícím způsobem:

- instance objektu Položka hierarchie nemůže vzniknout, aniž by existovala instance objektu Položka číselníku,
- pokud je instance objektu Položka číselníku navázána na konkrétní instanci objektu Položka hierarchie, nesmí být daná Položka číselníku v rámci dané Hierarchie číselníku použita víckrát,
- v případě, že by došlo ke smazání instance objektu Položka číselníku, musí být smazána i související instance objektu Položka hierarchie. V případě, že by došlo ke smazání instance objektu Položka hierarchie, není související instance objektu Položka číselníku nijak dotčena.

Objekt Položka hierarchie se váže sám na sebe (rekurzivní vazba; název vazby nadřízená položka):

- každá Položka hierarchie smí mít určenou maximálně jednu Položku hierarchie jako nadřízenou uzlovou položku. Nadřízenou položkou se myslí přímá nadřízená položka (nepřímých nadřízených položek smí mít jedna Položka hierarchie více jak jednu),
- v každé Hierarchii číselníku existuje právě jedna instance třídy Položka hierarchie, která nemá přímou nadřízenou Položku hierarchie. Taková položka se nazývá kořenová položka a v hierarchii položek stojí nejvýše a všechny ostatní Položky hierarchie jsou jí podřízené,
- vazba nadřízená položka není nijak omezená, co se týká počtu úrovní hierarchie. Zanoření Položek hierarchie je tak neomezeně hluboké,
- jedna Položka hierarchie smí být v rámci jedné Hierarchie číselníku použita maximálně jednou.

3.8.1 Atributy objektu Položka hierarchie

Objekt Položka hierarchie nemá standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů).

Objekt Položka hierarchie má definovány tyto atributy:

- **nadřízená položka**: tzv. odvozený atribut, který vznikne na základě nepovinné rekurzivní asociační vazby „nadřízená položka“ a pomocí kterého bude možno budovat hierarchii mezi Položkami číselníku a vytvářet tak hierarchii číselníkových položek.

3.9 Objekt Doména číselníku

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“).

Účelem objektu Doména číselníku je vymezit podmnožiny Položek číselníku nebo Položek hierarchie, které společně dohromady tvoří nějaký smysluplný celek. Například nad číselníkem zemí (předpokládejme, že obsahuje všechny země na světě) lze postavit doménu, která bude obsahovat jen evropské země. Smyslem takové domény je nabídnout uživateli pouze omezený výčet hodnot určitého Číselníku nebo Hierarchie číselníku.

Doména číselníku je objekt, který je vytvořen seskupením vybraných položek jednoho Číselníku. Doména číselníku se používá při definování obsahu dimenzionálního parametru. Doména je vytvořena buď prostým výběrem Položek číselníku anebo je odvozena

z Hierarchie číselníku, ale není přípustná kombinace výběru položek z Hierarchie číselníku a Číselníku současně do jedné Domény číselníku. Doména číselníku s vazbou na Hierarchii číselníku musí respektovat vztahy Položek číselníku určené touto Hierarchií číselníku.

Doména číselníku může být použita pro definování oboru Hodnot údajů jako základ formátových kontrol.

Pro jeden Číselník, resp. pro jeho položky, může být vytvořeno neomezeně Domén číselníku. Doména může být vytvořena nad globálním i lokálním Číselníkem.

V systému SDAT jsou rozeznávány následující typy domén:

a) z pohledu způsobu vytvoření:
   - **ruční:** doména, která je vytvořena buď výběrem položek přímo z Číselníku, nebo výběrem položek z určité Hierarchie číselníku, případně výběrem části stromu hierarchie daného číselníku. Může obsahovat jednu nebo více úrovní hierarchie, případně kombinaci úrovní hierarchie. Doménu vytváří Uživatel. Pokud je doména vytvořena z hierarchie, zachovává vždy její hierarchickou strukturu,
   - **automatická:** doména je určena pro použití v dynamických výkazech, aby nebylo možné kombinovat součtovou položku a její detail pro popis dynamického údaje (nejedná se o tzv. součtový řádek). Doména se generuje na pokyn uživatele pro konkrétní uzel hierarchie. Kód automatické domény generované z hierarchie má definována pravidla tak, aby tyto domény byly na první pohled odlišeny od ručních domén (např. A_“kód uzlové položky“_“pořadové číslo“). Automatická doména číselníku vytvořená z Hierarchie číselníku je vždy jednoúrovňová,

b) z pohledu aktualizace domény:
   - **doména aktualizovatelná s hierarchií:** doména vytvořená z Hierarchie číselníku, je automaticky systémem synchronizována s touto hierarchií, eviduje na ni vazbu a atribut, že se jedná o tento typ domény. Domény, které obsahují atribut „aktualizovatelná s hierarchií“, lze dále rozlišit na:
     - **s úplným výčtem (úplná):** taková doména, která obsahuje alespoň jeden úplný uzel hierarchie. Úplným uzlem lze chápat buď úplný výčet elementárních položek uzle, nebo úplný výčet některé úrovně hierarchie uzle, případně kombinaci těchto dvou možností (tj. jedná se o položky „v tom“). Při aktualizaci nebo synchronizaci Hierarchie číselníku se tato aktualizace promítá do odpovídajících Domén číselníku podle dohodnutých pravidel,
     - **s neúplným výčtem (neúplná):** doména, která neobsahuje ani jeden úplný výčet, tzn. obsahuje jen neúplné uzle hierarchie (tj. jedná se o položky „z toho“). Doménu číselníku s neúplným výčtem lze automaticky aktualizovat pouze v případě, že je z Hierarchie číselníku odebrána Položka hierarchie, která je obsažena v Doméně číselníku (v takovém případě je Položka hierarchie z Domény číselníku rovněž odebrána),
   - **doména neaktualizovaná s hierarchií:** doména, která je vytvořena přímo z výčtu Položek číselníku.

Atribut o úplnosti nebo neúplnosti výčtu v Doméně číselníku slouží k informování uživatele, jak budou generovány automatické kontroly. Níže jsou pro názornost uvedeny příklady úplnosti výčtu v doménách.

Hierarchie H1

```
H1
    S_A
        A1
        A2
    S_B
        B1
        B2
```

Doména D1 (úplný výčet z hierarchie H1)	Doména D2 (neúplný výčet z hierarchie H1 – z toho)	Doména D3 (úplný výčet z hierarchie H1)	Doména D4 (neúplný výčet z hierarchie H1 – z toho pro část B)
H1 S_A S_B	H1 A1 B1	H1 A1 A2 B1 B2	H1 S_A A1 A2 B1
Doména je úplná v kořenu H1	*Doména je neúplná*	*Doména je úplná v kořenu H1*	*Doména je úplná v uzlu S_A*

Tabulka 4 - Domény v hierarchii - příklady

Objekt Doména číselníku se váže k objektu Číselník a objektu Hierarchie číselníku takto:

- účelem této vazby je definovat, nad jakým Číselníkem a případně Hierarchií číselníku je Doména číselníku vybudována. Bude umožněno omezit při tvorbě Domény číselníku nabídku Položek číselníku nebo Položek hierarchie pouze na ty položky, které patří do jednoho Číselníku. V rámci jedné Domény číselníku nesmí být použity Položky číselníku nebo Položky hierarchie z více různých Číselníků nebo Hierarchií číselníku,
- Číselník nebo Hierarchie číselníku může vzniknout a trvale existovat, aniž by z něj byla vytvořena alespoň jedna instance objektu Doména číselníku. To znamená, že Číselník nebo Hierarchie číselníku jsou v systému užitečné i bez vytvoření instance Doména Číselníku,
- z jednoho Číselníku nebo Hierarchie číselníku může být vytvořeno N (neomezený) počet Domén číselníku.
- jedna Doména číselníku je vytvořena v rámci právě jednoho Číselníku nebo Hierarchie číselníku. Doména číselníku nemůže vzniknout, pokud neexistuje Číselník nebo Hierarchie číselníku,
- v případě, že dojde ke smazání Číselníku nebo Hierarchie číselníku, dojde ke smazání všech Domén číselníku, které jsou v rámci mazaného Číselníku nebo Hierarchie číselníku vytvořeny v případě, že systém nezjistí použití Domény v jiných objektech.

Aktualizace Domén číselníku, které vznikly z Hierarchie číselníku, je řízena následujícími pravidly (příklady k pravidlům jsou uvedeny v příloze 8.3 Příloha 3 - Příklady pro pravidla aktualizace Domén číselníku):

1. aktualizace se týká pouze Domén číselníku vytvořených z Hierarchie číselníku. Domén číselníku vytvořených vybráním položek z Číselníku se aktualizace nedotkne,
2. aktualizace se týká pouze úplných uzlů hierarchie u Domén číselníku vytvořených z Hierarchie číselníku. Neúplný uzel hierarchie zůstává změnou v Hierarchii číselníku nedotčen,
3. pokud je smazána nebo vkládána Položka hierarchie z/do hierarchie, projeví se tato změna i v Doméně číselníku vzniklé z této Hierarchie číselníku (podle pravidla 2),
4. pokud je smazána uzlová Položka hierarchie a její podřízené položky zůstávají, je změna promítnuta do Domény číselníku tak, že zůstávají jen elementární položky, aby se elementární obsah nadřízeného úplného uzle hierarchie nezměnil,
5. pokud je vkládána nová uzlová položka do Hierarchie číselníku, Doména číselníku vzniklá z této hierarchie zůstává nedotčena, může se změnit jen její atribut úplnosti,
6. při smazání Položky hierarchie z Hierarchie číselníku může dojít ke změně úplnosti uzlové položky. Z neúplné uzlové položky se může stát v rámci úpravy Hierarchie číselníku úplná uzlová položka,
7. při změně uzlové položky na elementární v Hierarchii číselníku, zůstává položka v Doméně číselníku, změna se projeví pouze v úplnosti,
8. zajištění úplnosti, tzn. doplnění elementárních položek úplné uzlové položky je prováděno v každé podřízené větvi na nejbližší nižší podřízené úrovni hierarchie Domény číselníku,
9. akce zajištění úplnosti bude prováděna:
    a. jako poslední akce při aktualizaci Hierarchie číselníku pro Domény číselníku vzniklé z aktualizované hierarchie,
    b. jako poslední akce při synchronizaci pro Domény číselníku vytvořené ze souvisejících Hierarchií číselníku,
10. pořadí položek v Doméně číselníku vytvořené z Hierarchie číselníku je shodné s pořadím položek Hierarchie číselníku.

3.9.1 Atributy objektu Doména číselníku

Objekt Doména číselníku obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů) s výjimkou následujícího:

- Garant.

Nad rámec standardních atributů jsou u objektu Položka hierarchie definovány následující atributy:

- **způsob aktualizace domény:** určuje, zda je doména vytvořena z Číselníku nebo z Hierarchie Číselníku a tento atribut přiděluje systém. Pokud je Doména číselníku vytvořena z Hierarchie číselníku, pak se provádí její aktualizace (změna obsahu) na základě definovaných pravidel. Tento atribut může nabývat stavů (H/ C), kdy
    - H = hierarchie: doména je aktualizována společně s hierarchií, ze které vznikla,
    - C = číselník: doména je vytvořena z položek Číselníku a neaktualizuje se,
- **typ Domény číselníku:** podle způsobu vytvoření instance objektu Doména číselníku nabývá atribut hodnot „automatická“ nebo „ruční“.

3.10 Objekt Převodník

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Účelem objektu Převodník je umožnit definovat vazby (převodní můstky) mezi položkami z právě dvou různých číselníků, resp. jejich položek.

Typickým případem, kdy bude využit objekt Převodník, bude situace, kdy budeme chtít udržet informaci o tom, v jaké zemi se platí jakou měnou:

- předpokládejme, že existuje číselník „Země“ (ISO3166), který obsahuje Položky číselníku ČR, SR, Německo, Maďarsko a existuje číselník „Měna“ (ISO4217), který obsahuje Položky číselníku EUR, CZK, HUF,
- založíme Převodník, kde definujeme jako „zdrojový číselník“ číselník „Země“ a jako „cílový číselník“ číselník „Měna“,
- následně dojde k definici vazeb (převodních můstků), které provedou mapování položek ze zdrojového číselníku na položky cílového číselníku, vzniknout tedy tyto vazby (vazby budou definovány v objektu Položka převodníku):
    - ČR = CZK,
    - SR = EUR,
    - SR = SKK,
    - Německo = EUR,
    - Maďarsko = HUF.

Objekt Převodník obsahuje právě dvě vazby na objekt Číselník. Pomocí těchto vazeb se určuje tzv. „zdrojový číselník“, jehož položky budou stát na „levé“ straně převodního můstku, a tzv. „cílový číselník“, jehož položky budou stát na „pravé“ straně převodního můstku. Není dovoleno, aby v rámci objektu Převodník byl jako zdrojový a cílový číselník použit vždy jeden a tentýž Číselník. Musí se vždy jednat o dva různé Číselníky.

Objekt Převodník se váže na objekt Položka převodníku:

- Převodník může vzniknout, aniž by obsahoval alespoň jednu Položku převodníku. Trvalá existence objektu Převodník bez souvisejících instancí objektu Položka převodníku je objektově možná, ale z hlediska smysluplnosti systému nemá význam,
- Převodník může obsahovat N (neomezeně) Položek převodníku,
- pokud vzniká instance objektu Položka převodníku, pak musí být přiřazena k právě jedné instanci objektu Převodník,
- pokud dojde ke smazání instance objektu Převodník, dojde ke smazání všech souvisejících instancí objektu Položka převodníku.

3.10.1 Atributy objektu Převodník

Objekt Převodník obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů).

Nad rámec standardních atributů nejsou u objektu Převodník definovány žádné další atributy.

3.11 Objekt Položka převodníku

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Účelem objektu Položka převodníku je umožnit definovat vazby (převodní můstky) mezi položkami z právě dvou různých číselníků, resp. jejich položek. Jedná se o objekt, který bezprostředně souvisí s objektem Převodník a podrobný popis účelu je tak uveden v kapitole 3.10. Objekt Převodník.

Položka převodníku má právě dvě vazby na objekt Položka číselníku. V rámci každé vazby je definována jedna část převodního můstku. Existuje tak vazba s názvem „zdrojová položka“ a „cílová položka“. Instance objektu Položka převodníku je kompletní tehdy, pokud obsahuje právě jeden odkaz na Položku číselníku definovaný jako „zdrojová položka“ a právě jeden odkaz na Položku číselníku definovaný jako „cílová položka“.

Pro objekt Položka číselníku platí:

- jedna Položka číselníku může být použita ve více Položkách převodníku, ale „zdrojová položka“ nebo „cílová položka“ má vazbu právě na jednu Položku číselníku. To znamená, že pokud bude třeba vytvořit převodník mezi Položkami číselníku Měna a Země, je možno dosáhnout nastavení EUR – Německo, EUR – Slovensko a EUR – Francie (jedna Položka číselníku (EUR) je použita víckrát), ale z hlediska objektového modelu se jedná o tři různé instance objektu Položka převodníku,
- v rámci definice „zdrojové položky“ mohou být použity pouze ty Položky číselníku z Číselníku, který je v rámci nadřízeného objektu Převodník definován jako „zdrojový číselník“,
- v rámci definice „cílové položky“ mohou být použity pouze ty Položky číselníku z Číselníku, který je v rámci nadřízeného objektu Převodník definován jako „cílový číselník“,
- pokud by došlo ke smazání instance Položka číselníku, pak musí dojít ke smazání všech instancí objektu Položka převodníku, ať už se mazaná Položka číselníku objevuje v dané instanci jako „zdrojová položka“ nebo „cílová položka“.
- není povolená existence duplicitní kombinace „zdrojová položka“ – „cílová položka“ v rámci jednoho Převodníku. Kombinace „zdrojová položka“ – „cílová položka“ se v rámci jednoho Převodníku smí objevit maximálně jednou.

3.12 Objekt Účtová osnova

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Instance objektu Účtová osnova vzniká bez vazby na jakýkoli jiný objekt nebo instanci objektu. Účelem tohoto objektu je zastřešit existenci různých druhů účtových tříd a účtových skupin pro různé skupiny Vykazujících osob. V systému tak může existovat N (neomezeně) Účtových osnov. Neomezený počet Účtových osnov je třeba proto, aby bylo možno v systému podchytit účtové rozvrhy (seznamy účtů), které jsou specifické pro různá odvětví Vykazujících osob. Odlišné Účtové osnovy se tak budou používat například pro banky, pojišťovny, investiční společnosti, nefinanční podniky apod.

Vazba instance objektu Účtová osnova na objekt Ukazatel je volitelná (nemusí být definována) a je daná typem vazby (např. debetní zůstatek, kreditní zůstatek), rozsahem vazby (např. celý účet, nebo jeho část) a možností opačné hodnoty. Tato vazba se definuje pomocí objektu Zařazení účtu k ukazateli. Při definici vazby instance objektu Účtová osnova na objekt Ukazatel lze používat všechny tři úrovně hierarchie.

Objekt Účtová osnova (a podřízený objekt Účet) je v systému zaveden proto, aby bylo možno lépe popsat objekt Ukazatel, který Vykazující osobě jasně vymezuje, jaká data mají být dodána (za pomoci přesné definice účtů z účtového rozvrhu Vykazující osoby může pak Vykazující osoba získat požadovaný Údaj (popsaný Ukazatelem) přímo ze svého účetnictví).

Objekt Účtová osnova se váže na objekt Účet:

- Účtová osnova může vzniknout a trvale existovat, aniž by obsahovala alespoň jeden Účet. Existence instance objektu Účtová osnova bez podřízených instancí objektu Účet nedává metodický smysl,
- Účtová osnova může obsahovat N (neomezeně) Účtů,
- pokud vzniká Účet, pak musí být přiřazen k právě jedné Účtové osnově,
- konkrétní Účet je přiřazen právě jedné Účtové osnově (neexistuje sdílení Účtů, pokud je třeba nějaký Účet mít v různých Účtových osnovách, je tento Účet vytvořen znovu a je tak v systému redundantně),
- v případě potřeby změnit zařazení Účtu do jiné Účtové osnovy, dojde k přepsání informace o tom, do jaké Účtové osnovy byl Účet zařazen (Účet „UCET1“ má původně vazbu na Účtovou osnovu „UO1“ a po změně má vazbu na „UO2“, přičemž dojde ke ztrátě informace, že v minulosti měl Účet „UCET1“ vazbu na Účtovou osnovu „UO1“),
- pokud dojde ke smazání instance objektu Účtová osnova, dojde ke smazání všech souvisejících instancí objektu Účet.

3.12.1 Atributy objektu Účtová osnova

Objekt Účtová osnova obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů).

Nad rámec standardních atributů nejsou u objektu Účtová osnova definovány žádné další atributy.

3.13 Objekt Účet

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Objekt Účet je podřízený objektu Účtová osnova a podrobněji je vazba na tento objekt popsána výše. Účelem tohoto objektu je vytvořit seznam účtů, které jsou společné v rámci jedné účtové osnovy. Objekt Účet je definován jako speciální hierarchický seznam jednotlivých položek účtové osnovy. V rámci objektu Účet existují tři různé úrovně účtů. Tyto typy účtů jsou mezi sebou navzájem striktně propojeny a platí:

- jedna Účtová osnova může obsahovat N (neomezeně) Účtů nejvyšší úrovně (neplatí tak, že by Účtová osnova měla právě jednu kořenovou položku tak, jak je to obvyklé u Hierarchií číselníku). Typ účtu umístěný na nejvyšší úrovni se nazývá „účtová třída“,
- v rámci jedné účtové třídy se může vyskytovat N (neomezeně; to znamená, že i žádný) Účtů druhé úrovně. Takové účty nazýváme „účtová skupina“. Účtová skupina dále konkretizuje účtovou třídu. Účtová skupina nemůže vzniknout, aniž by byla přiřazena k právě jedné účtové třídě,
- v rámci jedné účtové skupiny se může vyskytovat N (neomezeně; to znamená, že i žádný) Účtů třetí úrovně. Takové účty nazýváme „syntetický účet“. Syntetický účet dále konkretizuje účtovou skupinu. Syntetický účet nemůže vzniknout, aniž by byl přiřazen k právě jedné účtové skupině.

Výše uvedený princip je v modelu realizován tak, že objekt Účet udržuje vazbu sám na sebe. Tato vazba se nazývá „nadřízený účet“ a lze pomocí ní vytvořit neomezeně hlubokou hierarchii účtů. I když je tato hierarchie z principu neomezená, na aplikační úrovni bude omezena na maximálně tři úrovně.

Objekt Účet se (přes asociační třídu Zařazení účtu k ukazateli) váže na objekt Ukazatel. Na této vazbě jsou dodatečné atributy, které dále specifikují přiřazení konkrétního účtu k Ukazateli, přičemž platí:

- Ukazatel může být popsán neomezeným počtem Účtů (a platí, že nemusí být popsán pomocí Účtů vůbec), nicméně platí omezení, že tyto Účty musejí být právě z jedné Účtové osnovy,
- jeden Účet může popisovat neomezený počet Ukazatelů, nicméně platí omezení, že v rámci popisu konkrétního Ukazatele smí být použit konkrétní Účet maximálně jednou,
- Ukazatel smí být popsán jakýmkoli Účtem, tj. k popisu může být použita jakákoli úroveň hierarchie účtu, tedy Ukazatel může být popsán buď konkrétním syntetickým účtem, ale i účtovou skupinou a účtovou třídou. V případě, že je Ukazatel popsán vyšší než třetí úrovní (syntetický účet), má se za to, že tento popis znamená „všechny podřízené syntetické účty“ (pokud není stanovena výjimka, viz dále),
- v rámci propojení objektů Účet a Ukazatel je třeba povinně definovat následující atributy:
    - **znaménko:** určuje, jakým způsobem je hodnota vykázána v účetnictví Vykazující osoby započtena v rámci daného Ukazatele. Jsou přípustné pouze dvě možnosti: „plus“ nebo „minus“,
    - **způsob zařazení:** určuje, jakým způsobem je Účet k Ukazateli zařazen. Přípustné jsou dvě možnosti „include“ a „exclude“, kde „include“ znamená, že se přiřazený Účet k Ukazateli započítává, a „exclude“znamená, že má být ze započtení vynechán. Tento institut umožňuje definovat princip tzv. negativních výjimek (například přiřazením Účtu "17 - Závazky z cenných papírů" (include), čímž je řečeno, že je vybráno "vše, co je vykázáno na syntetických účtech v této účtové skupině", a následně je přidán Účet "174 - Závazky z akcií" (exclude), čímž je řečeno "všechno, co je na syntetických účtech ve skupině 17, a zároveň nezapočítávat to, co je na syntetickém účtu 174),
    - **typ zařazení:** určuje, jaká část účtu má být do popisu zahrnuta, přípustné jsou tyto hodnoty:
        - kreditní obrat,
        - debetní obrat,
        - výsledný kreditní obrat,

- výsledný debetní obrat,
- výsledný obrat,
- kreditní nebo debetní obrat,
- kreditní zůstatek,
- debetní zůstatek,
- výsledný kreditní zůstatek,
- výsledný debetní zůstatek,
- výsledný zůstatek,

- **rozsah účtu:** určuje, zda si osoba, která předává data, může určit, zda pro vykazování použije pouze některé analytické účty z předepsaného syntetického účtu anebo musí použít celý syntetický účet. Jsou povoleny tyto hodnoty:
  - celý SY účet.
  - část SY účtu (analytické účty).

3.13.1 Atributy objektu Účet

Objekt Účet obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů) s výjimkou následujícího:

- Garant.

Nad rámec standardních atributů obsahuje objekt Účet následující atributy:

- **charakter účtu**: může nabývat tří hodnot: aktivní, pasívní, aktivní i pasívní,
- **nadřízený účet**: odvozený atribut, který vznikne na základě nepovinné rekurzivní asociační vazby „nadřízený účet“ a pomocí kterého bude možno budovat hierarchii mezi účty.

3.14 Objekt Datový typ

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Účelem tohoto objektu je nadefinovat množinu základních datových typů, které je možno v rámci systému používat při popisování Údajů. Instance objektu Datový typ vzniká bez vazby na jakýkoli jiný objekt, či instanci objektu. Nejedná se o klasické datové typy jako například VARCHAR/CHAR/DATE/NUMBER, ale o jejich rozšíření (výše uvedené datové typy jsou instanci objektu Datový typ přiřazeny jako atribut). Je tak možno vytvořit Datový typ „DAT1“, který je postaven nad datovým typem VARCHAR (právě jedním), ale dále jej rozšiřuje, například informací o délce a masce. Příkladem instance objektu Datový typ je například datový typ „ISIN“. Tento datový typ je popsán takto:

- Elementární datový typ = VARCHAR
- Délka = 12
- Maska = XXXXXXXXXXXX (12 po sobě jdoucích písmen nebo čísel).

Objekt Datový typ se váže na objekt Doména datového typu. Popis vazby je uveden v kapitole 3.15 Objekt Doména datového typu. Dalšími atributy tohoto objektu bude „násobek“ a „regulární výraz“.

Od Datového typu může být odvozeno 0 až N Domén datového typu. Datový typ může být v systému použit vícenásobně. Datový typ může být použit pro definování oboru hodnot údajů jako základ formátových kontrol.

U každého Datového typu je možné:

- určit jeho masku, délku, násobek a případně definovat typ jednotky (%, měna, násobek, formát data),
- dospecifikovat masku údajů tak, aby rozlišovala malá a velká písmena, kombinaci číslic a písmen, možnost určit počet povinně vyplňovaných míst,
- kontrolovat správnost vyplnění určitých Datových typů (např. rodného čísla, IČ, BIC aj.),
- definovat kladné nebo záporné hodnoty s přesným počtem definovaných desetinných míst,
- určit regulární výraz, pomocí kterého se bude validovat vstup uživatele.

3.14.1 Atributy objektu Datový typ

Objekt Datový typ obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů).

Nad rámec standardních atributů nejsou u objektu Datový typ definovány žádné další atributy.

3.15 Objekt Doména datového typu

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Účelem objektu Doména datového typu je podrobnější specifikace instance objektu Datový typ. Pokud například bude definován datový typ N3 (elementární datový typ = NUMBER, délka = 3 čísla před desetinnou čárkou, 0 míst za desetinnou čárkou), pak k takovému datovému typu mohou být například definovány tyto domény:

- DOMENA1 – přípustné hodnoty intervalu jsou 0-100,
- DOMENA2 – přípustné hodnoty intervalu jsou 101-999.

Tímto zápisem (a následným použitím v systému) je umožněna podrobnější kontrola vykazovaných dat. Tam, kde uživatel řekne, že chce zpět dostat číslo v rozsahu 0-100, použije při projektování Doménu datového typu DOMENA1 a tam, kde chce zpět dostat číslo v rozsahu 101-999, použije Doménu datového typu DOMENA2. Pokud nechce obor hodnot nijak omezovat, nepoužije žádnou doménu, ale pouze datový typ N3, který sám o sobě zajišťuje rozsah hodnot 0-999, tedy maximálně 3 celá kladná čísla.

Vazba objektu Doména datového typu na objekt Datový typ je postavena takto:

- Instance objektu Doména datového typu nemůže vzniknout bez existence instance Datový typ,
- pokud vzniká instance objektu Doména datového typu, pak vzniká právě nad jednou instancí objektu Datový typ,
- Datový typ může vzniknout (a existovat), aniž by byla vytvořena vazba na Doménu datového typu,

- v případě, že by došlo ke smazání instance třídy Datový typ, budou smazány i všechny související instance třídy Doména datového typu.

3.15.1 Atributy objektu Doména datového typu

Objekt Doména datového typu obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů).

Nad rámec standardních atributů nejsou u objektu Doména datového typu definovány žádné další atributy.

3.16 Objekt Ukazatel

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Účelem objektu Ukazatel je základní charakteristika (popis) údaje, který potřebuje ČNB získat od Vykazujících osob. Jedná se tedy o definici toho, jaké Hodnoty údajů má Vykazující osoba dodat, aby byla splněna její Vykazovací povinnost. Obsah objektu Ukazatel si tak můžeme představit jako seznam položek, které jsou předmětem vykazování. Ukazatel je tak například:

- Aktiva celkem,
- Pokladní hotovost,
- Pohledávky,
- Pozemky, budovy, zařízení,
- Nehmotný majetek.

Objekt Ukazatel sám o sobě eviduje minimum informací nutných proto, aby byl Údaj popsán tak, aby mu Vykazující osoby rozuměly. Ukazatel tak udržuje především kód, název a popis ukazatele, samotná definice Ukazatele je realizována formou vazeb na jiné objekty systému.

Vazba objektu Ukazatel na objekt Účet je popsána pomocí odkazu na syntetické účty konkrétní Účtové osnovy tak, aby Vykazující osoba mohla přímo zjistit požadovaný údaj ze svého účetnictví. Podrobněji je vazba Ukazatel/Účet popsána v kapitole 3.13. Objekt Účet.

Instance objektu Ukazatel lze vytvářet jako hierarchické s tím, že jsou povoleny maximálně dvě úrovně této hierarchie. Hierarchie je zajištěna rekurzivní asociační vazbou objektu Ukazatel sám na sebe s názvem „Ukazatel druhé úrovně“ Rozlišujeme tak:

- Ukazatel první úrovně – Jedná se o instanci objektu Ukazatel, která nemá vazbu na jinou instanci objektu Ukazatel.
- Ukazatel druhé úrovně– Jedná se o instanci objektu Ukazatel, která má vazbu na právě jednu další instanci objektu Ukazatel, a to vždy takovou, která je první úrovně. Tato vazba je neměnná.

Správa ukazatelů probíhá v knihovně.

Pro instance objektu Ukazatel platí tato pravidla:

- Každý Ukazatel první úrovně může mít 0 až N ukazatelů druhé úrovně. Pro ukazatel první úrovně platí, že je povoleno, aby existoval bez tzv. dodatečné konkretizace, tedy bez

napojení na objekt Parametr, nicméně je zároveň povoleno, aby tato vazba existovala. V případě, že Ukazatel první úrovně je napojen na parametr, pak platí:
    - Vzniká instance v asociační třídě Dodatečná konkretizace Ukazatele (viz další text).
    - Jakýkoli další Ukazatel druhé úrovně automaticky přebírá konkretizaci Ukazatele první úrovně (není třeba ji již znovu definovat). Tuto konkretizaci nejde Ukazatelům druhé úrovně odebrat. Modifikovat ji lze pouze na Ukazateli první úrovně, od kterého všechny Ukazatele druhé úrovně modifikovanou konkretizaci opětovně zdědí.
- Každý Ukazatel první úrovně může být navázán na 0 až N Účtů. Pro Ukazatel první úrovně platí, že je povoleno, aby existoval bez navázaných Účtů, nicméně je povoleno, aby tato vazba existovala. V případě, že Ukazatel první úrovně je napojen na Účet, pak platí:
    - Vzniká instance v asociační třídě Zařazení Účtu k Ukazateli.
    - Jakýkoli další Ukazatel druhé úrovně automaticky přebírá navázané Účty Ukazatele první úrovně (není třeba je již znovu navazovat). Navázané účty lze dále odebírat či modifikovat bez vlivu na Ukazatele první úrovně.
- Ukazatel druhé úrovně musí být podřízen právě jednomu Ukazateli první úrovně. Ukazatel druhé úrovně tak přebírá konkretizaci nebo navázané Účty od rodičovského Ukazatele (Ukazatel první úrovně), pokud tato konkretizace či navázání na Účty existuje. Zároveň pro Ukazatel druhé úrovně platí:
    - Musí obsahovat alespoň jednu další dodatečnou konkretizaci, tedy napojení na alespoň jednu další instanaci objektu Parametr přes asociační třídu Dodatečná konkretizace Ukazatele.
    - Pokud obsahuje dodatečnou konkretizaci, která již existuje (napojení na stejnou množinu parametrů jako jiný ukazatel druhé úrovně, který je napojen na stejný ukazatel první úrovně), musí se tento Ukazatel druhé úrovně lišit ve vazbě na Účty oproti shodně dodatečně konkretizovanému Ukazateli druhé úrovně.
- Kód Ukazatele druhé úrovně bude odvozen z kódu Ukazatele první úrovně a bude jednoznačný v rámci celého systému.
- Název Ukazatele bude jednoznačný v rámci celého systému.
- Systém zajistí, že nebude možné vytvořit stejně definovaný Ukazatel druhé úrovně a to v žádném ze stavů. Stejnou definicí Ukazatele druhé úrovně se rozumí stejná kombinace Ukazatele první úroveň, stejná sada přiřazených parametrů (Dodatečná konkretizace ukazatele) a stejná vazba na Účty.

3.16.1 Objekt Dodatečná konkretizace Ukazatele

Primárně jsou všechny Ukazatele zařazené do Datové oblasti konkretizovány parametry, které platí pro celou datovou oblast (tzv. „společné parametry“). To je zajištěno objekty „Parametr“ a „Vykazovaný parametr“. Tyto společné parametry se tak vztahují ke konkrétnímu výkazu/datové oblasti.

Dodatečná konkretizace Ukazatele má zajistit, aby jeden Ukazatel mohl být konkretizován dalšími (dodatečnými) Parametry, které však budou určeny speciálně pro jeden Ukazatel a v datové oblasti tak vytvoří takzvané „nespolečné parametry“, tedy parametry, které neplatí pro celou datovou oblast, ale pouze pro ty Ukazatele, ke kterým se váží. Nespolečné parametry jsou k Ukazatelům přiřazovány již v knihovně.

Asociační třída „Dodatečná konkretizace ukazatele“ spojuje dva objekty knihovny, objekt Ukazatel a objekt Parametr. V případě, že je Ukazatel v knihovně napojen na nějaký parametr, pak je v rámci tohoto napojení navíc ještě potřeba definovat, jako položkou číselníku je tato vazba dodatečně popsána (jak je provedena konkretizace), případně na jakou doménu datového typu. Příkladem může být dodatečná konkretizace Ukazatele „Aktiva dlouhodobá“. Jedná se o ukazatel druhé úrovně, který bude podřízen ukazateli první úrovně s názvem „Aktiva“. Protože se jedná o ukazatel druhé úrovně, musí být parametrizován alespoň jedním parametrem (navíc takovým, kterým není popsán jeho rodič) – v tomto případě to bude parametr „P0065 – Splatnost“, na který bude navázána číselníková položka „Dlouhodobá“ z číselníku „Doba splatnosti“. V asociační třídě „Dodatečná konkretizace ukazatele“ tak budou minimálně uloženy tyto identifikátory:

- ID Ukazatele „Aktiva dlouhodobá“
- ID Parametru „P0065 - Splatnost“
- ID Číselníkové položky „Dlouhodobá“

Vzhledem k tomu, že se stále jedná o objekt knihovny, lze takový Ukazatel používat pro projektování v libovolném množství výkazů. Zároveň je ale umožněno, aby v každém výkazu byl daný Ukazatel použit s jiným oborem hodnot, tedy například aby v jednom výkazu byly vykázány data v tisicích, v jiném v miliónech apod. Protože se však ale jedná o dodatečnou informaci, která závisí na použití Ukazatele v konkrétním výkazu, pak je tento obor hodnot třeba definovat až ve vazbě konkrétního ukazatele na konkrétní datovou oblast. Za tím účelem je v systému zaveden objekt „Ukazatel v Datové oblasti“.

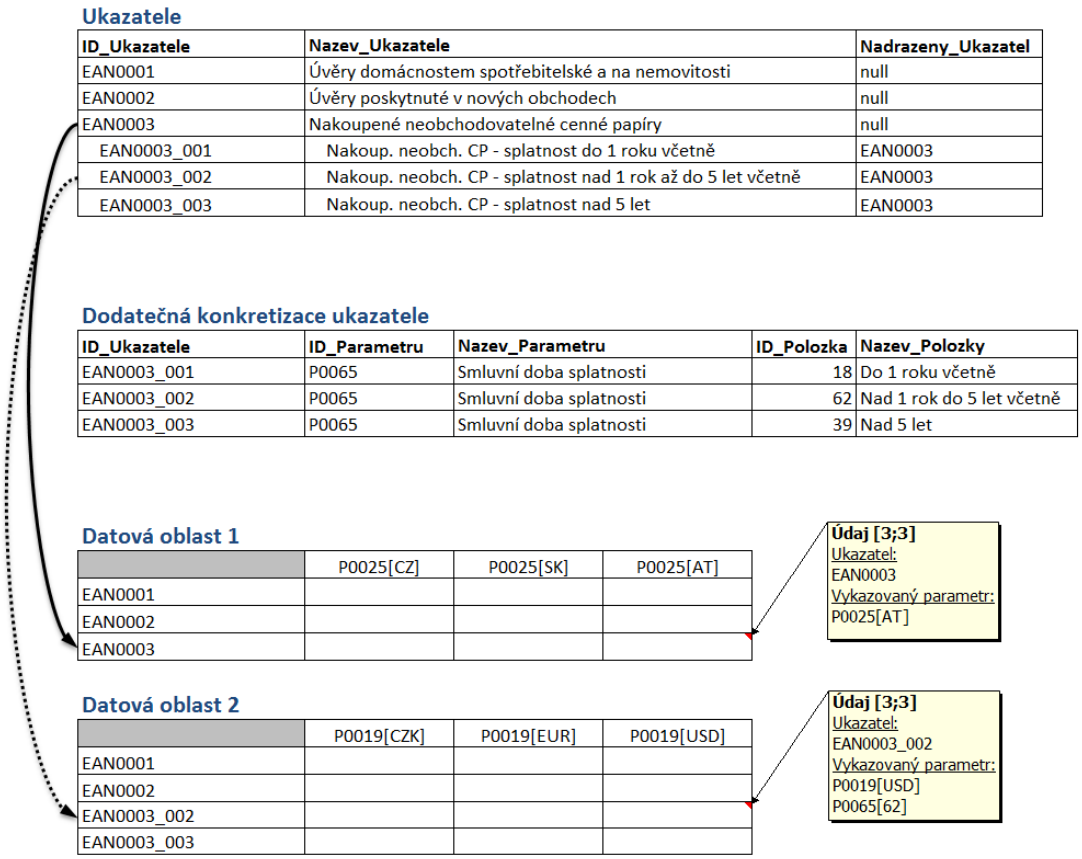
Pro dodatečnou konkretizaci Ukazatele v knihovně platí:

- jeden Parametr smí dodatečně konkretizovat více Ukazatelů. Jeden Parametr smí být použit pro daný ukazatel ukazatel maximálně jednou.
- Ukazatel jakékoli úrovně bude moci mít konkretizované parametry na konstantní hodnotu.
  - Ukazatel první úrovně bude mít tuto konkretizaci nepovinnou.
  - Ukazatel druhé úrovně bude mít tuto konkretizaci povinnou.
  - Pokud bude mít Ukazatel první úrovně dodatečně konkretizované parametry, mají podřízené Ukazatele druhé úrovně tuto dodatečnou konkretizaci daných parametrů shodnou.
- V případě, že uživatel bude měnit dodatečnou konkretizaci Ukazatele druhé úrovně, pak v případě, že tento měněný Ukazatel druhé úrovně
  - ještě nebyl nikde použit (nebyl připojen k žádné Datové oblasti), NEVZNIKNE nový Ukazatel druhé úrovně, ale změní se dodatečná konkretizace existujícího Ukazatele.
  - je někde použit (je již připojen k alespoň jedné Datové oblasti), pak:
    - Systém založí novou instanci objektu Ukazatel s tím, že původní Ukazatel druhé úrovně ponechá nezměněný.

- Systém zobrazí uživateli informaci o tom, v jakých Datových oblastech je měněný Ukazatel druhé úrovně použit a umožní uživateli určit, v jakých Datových oblastech má ponechat původní Ukazatel druhé úrovně a v jakých má použít nově vzniklý Ukazatel druhé úrovně.

Příklad:

- V knihovně jsou vytvořené Ukazatele první úrovně
  - „EAN0001“,
  - „EAN0002“
  - „EAN0003“, tento ukazatel má tyto Ukazatele druhé úrovně:
    - „EAN0003_001“,
    - „EAN0003_002“,
    - „EAN0003_003“ (viz Obrázek 12 – Dodatečná konkretizace ukazatele).
- Žádný z Ukazatelů („EAN0001“, „EAN0002“ a „EAN0003“) nemá, ale může mít související instanci v objektu „Dodatečná konkretizace ukazatele“.
- Každý z Ukazatelů druhé úrovně („EAN0003_001“, „EAN0003_002“ a „EAN0003_003“) má a musí mít související instanci v objektu „Dodatečná konkretizace ukazatele“.
- Každý Údaj odvozený z Ukazatele první úrovně je parametrizován pouze Parametry Datové oblasti (viz Datová oblast 1 – Údaj [3;3]).
- Každý Údaj odvozený z Ukazatele druhé úrovně je parametrizován Parametry Datové oblasti spolu s dodatečnými (nespolečnými) Parametry daného Ukazatele druhé úrovně (viz Datová oblast 2 – Údaj [3;3]).
- Zařazením jakéhokoli Ukazatele do Datové oblasti vzniká instance objektu Ukazatel v Datové oblasti.
- Údaje odvozené z Ukazatele EAN0003_002 budou mít vždy společný Parametr P0065 s položkou „62“. Další Parametry Údajů odvozených od tohoto Ukazatele se mohou v různých Datových oblastech lišit.



Obrázek 12 – Dodatečná konkretizace ukazatele

3.16.2 Objekt Ukazatel v Datové oblasti

Jedná se o asociační třídu, která spojuje objekt Ukazatel (Knihovna) a konkrétní Datovou oblastí (Výkaz). Platí, že každý Ukazatel, který je zařazen do Datové oblasti musí mít definován tzv. obor hodnot. Obor hodnot je možno definovat buď:

- Datový typem
- Doménou (doménou číselníku/datového typu)

Uživatel musí z výše uvedených možností vybrat právě jednu možnost; to znamená, že nemůže nechat Ukazatel zařazený do Datové oblasti bez toho, aniž by obor hodnot definoval (lze uvažovat o tom, že implicitně systém při zařazování Ukazatele do Datové oblasti nastaví každému Ukazateli implicitní obor hodnot (definován pomocí systémové proměnné). V asociační třídě Ukazatel v Datové oblasti budou minimálně uloženy tyto identifikátory:

- ID Ukazatele
- ID Datové oblasti
- ID Datového Typu/ID Domény číselníku

Asociační třída Ukazatel v datové oblasti tak udržuje nepovinnou asociační vazbu na objekty „Doména“ (objekt Doména je předek objektů „Doména číselníku“ a „Doména datového typu“; Instance objektu „Ukazatel v datové oblasti“ musí referovat na jednoho z těchto

potomků objektu „Doména“) a na objekt „Datový typ“. Korektně vytvořená instance objektu Ukazatel v datové oblasti musí splnit podmínku XOR (musí referovat na právě jednu instanci výše zmíněných objektů).

Platí tato pravidla:

- existence instancí obou objektů (jak objektu Ukazatel, tak objektu Datová oblast) v systému je zcela nezávislá, tedy jak instance objektu Datová oblast, tak instance objektu Ukazatel mohou v systému vzniknout samy (bez vazby na druhý objekt),
- jeden Ukazatel smí být zařazen do více Datových oblastí,
- jeden Ukazatel smí být zařazen vícekrát do jedné Datové oblasti a to dokonce na stejnou osu, je-li daná osa rozdělena do částí.
- Do datové oblasti nesmí být zařazen takový Ukazatel, který je v knihovně dodatečně konkretizován stejnou sadou parametrů (tzv. nespolečné parametry), jaká je přiřazena přímo k datové oblasti (tzv. společné parametry).
- Systém umožňuje jakýkoli ukazatel zařadit do datové oblasti a tam jej dodatečně konkretizovat, to ale nemění konkretizaci tohoto Ukazatele, nýbrž:
    - změní vazbu na již existující Ukazatel s takovou konkretizací (pokud existuje),
    - založí v Knihovně Ukazatel 2. úrovně s takovou konkretizací, jaká byla uživatelem nadefinovaná v rámci zařazení Ukazatele do Datové oblasti.
- jedna Datová oblast smí obsahovat více Ukazatelů,
- do jedné Datové oblasti nesmí být zařazena instance objektu Ukazatel první úrovně a zároveň instance objektu Ukazatele druhé úrovně, který je jeho potomkem.
- instance objektu Ukazatel v DO má následující pravidla:
    - musí být definován atribut „pořadí“, který určuje pořadí zařazení Ukazatele v rámci Datové oblasti; tato informace bude použita pro prezentaci dat (pro vygenerování "gridu" údajů v rámci Datové oblasti),
    - musí být definován atribut „povinnost“ (může nabývat hodnot „ano“ nebo „ne“ s tím, že defaultně je vytváření instance objektu Ukazatel v DO tento atribut nastaven na hodnotu „ne“). Pomocí tohoto atributu je definováno, zda k Údaji, který v budoucnu z daného Ukazatele vznikne, bude povinné zaslat nějakou hodnotu nebo ne.
    - musí být definován atribut „osa“, která určuje, na které ose v GRIDu Datové oblasti bude Ukazatel vykreslen. Atribut „osa“ může nabývat hodnot „X“ nebo „Y“ nebo „Z“.
    - Ukazatel může být zařazen pod jiný, do Datové oblasti již přiřazený, Ukazatel. Uživatel tak může vytvořit neomezenou hierarchii Ukazatelů v Datové oblasti (o uchování informace o nadřízeném Ukazateli se stará nepovinná asociační rekurzivní vazba „Nadřízený Ukazatel v DO“), tj.:
        - pokud vzniká hierarchie Ukazatelů, pak uživatel pro každý Ukazatel, který obsahuje nějaké podřízené Ukazatele, je třeba definovat, zda výčet podřízených Ukazatelů je úplný (tedy zda součet hodnot v podřízených ukazatelích se musí rovnat hodnotě ukazatele nadřízeného). Tato informace je uchována pomocí atributu „suma“. Na základě hodnoty tohoto atributu jsou pak automaticky generovány kontroly podle následujícího pravidla:
            - pro každý Ukazatel, který obsahuje nějaké podřízené Ukazatele a u kterého je uvedena v atributu „suma“ hodnota „ano“, bude vygenerována kontrola, která bude splněna tehdy, pokud suma

hodnot vykázaných na Ukazatelích, které jsou přímými potomky nadřízeného Ukazatele, **se rovnají** hodnotě vykázané v nadřízeném Ukazateli,

- pro každý Ukazatel, který obsahuje nějaké podřízené Ukazatele a u kterého je uvedena v atributu „suma“ hodnota „ne“, bude vygenerována kontrola, která bude splněna tehdy, pokud suma hodnot vykázaných na Ukazatelích, které jsou přímými potomky nadřízeného Ukazatele, **je menší nebo rovna** hodnotě vykázané v nadřízeném Ukazateli.

3.16.3 Atributy objektu Ukazatel

Objekt Ukazatel obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů) s výjimkou následujících:

- Garant.

Nad rámec standardních atributů obsahuje objekt Ukazatel tyto atributy:

- **Datatyp:** obsahuje informaci o tom, jakého typu hodnot může Ukazatel nabývat. Volba oboru hodnot v Datových oblastech se odvíjí od nastaveného datatypu.

3.17 Objekt Parametr

Sledování historie objektu: Sledování historie – časová platnost + stavy (viz kapitola 2.2.6 Přístup „Sledování historie – časová platnost + stavy“)

Účelem objektu Parametr je dodatečná charakteristika (popis) objektu Ukazatel. Objekt Parametr si tak můžeme představit jako množinu číselníkových položek, které nějakým způsobem detailně popisují Ukazatel. Příkladem Parametrů je například:

- **ISIN** (z něj následně vznikne Konkretizovaný parametr s vazbou na Datový typ),
- **Název cenného papíru** (z něj následně vznikne Konkretizovaný parametr s vazbou na datový typ),
- **Okres v ČR** (z něj následně vznikne Konkretizovaný parametr s vazbou na Číselník/Doménu číselníku),
- **Splatnost** (z něj následně vznikne Konkretizovaný parametr s vazbou na Číselník/Doménu číselníku),
- **Měna** (z něj následně vznikne Konkretizovaný parametr s vazbou na Číselník/Doménu číselníku),
- **Rozsah vykazování** (z něj následně vznikne Konkretizovaný parametr s vazbou na Číselník/Doménu číselníku).

Objekt Parametr se váže na objekt Číselník nebo Datový typ (XOR: právě jeden z nich, povinně). Tím se provede základní vymezení očekávané hodnoty zaslaného údaje. Podrobná specifikace omezení zasílaných údajů je pak následně definována v rámci konkretizace parametru (viz další text a popis objektu Konkretizace parametru níže).

Objekt Parametr se dále váže na objekt Datová oblast. Připojením Parametru k Datové oblasti vzniká sada tzv. „společných“ Parametrů. Tyto Parametry budou použity pro popis **všech**

údajů zařazených do Datové oblasti. Existence obou objektů (Parametr a Datová oblast) v systému je zcela nezávislá, tedy jak objekt Datová oblast, tak objekt Parametr mohou v systému vzniknout samy (bez vazby na druhý objekt). Dále platí:

- jeden Parametr smí být zařazen do více Datových oblastí,
- jedna Datová oblast smí obsahovat více Parametrů,
- jeden Parametr nesmí být vložen současně na více osách (osa „X“, „Y“ nebo „Z“) jedné Datové oblasti,
- pořadí Parametrů na jednotlivých osách definuje zároveň pořadí popisných pater (textů) ve struktuře Datové oblasti. Toto pořadí je možné změnit.
- v rámci vazby mezi Datovou oblastí a Parametrem vzniká instance objektu „Konkretizovaný parametr“, která dále specifikuje vazbu mezi instancí objektu Parametr a Datová oblast, a to takto:
    - každý Parametr zařazený do Datové oblasti musí mít právě jednu konkretizaci. Tedy pro každou kombinaci Parametr/Datová oblast musí vzniknout právě jedna instance objektu Konkretizovaný parametr,
    - pokud vzniká instance třídy Konkretizovaný parametr, pak:
        - musí tato instance mít přiřazen právě jeden z následujících objektů (tím vlastně vzniká ona konkretizace)
            - Položka (číselníku nebo hierarchie číselníku),
            - Hierarchie číselníku,
            - Doména číselníku (z položek číselníku nebo hierarchie), nebo Doména datového typu,
            - Datový typ.

V systému SDAT rozeznáváme následující typy parametrů:

a) z hlediska typu parametru
    - **popisný:** podle potřeby projektování lze definovat popisný parametr, který má vazbu na konkrétní Číselník nebo na Datový typ,
    - **identifikační:** parametr, který není součástí popisu Údaje, ale má význam pro jednoznačnou identifikaci zaslaných řádků dynamické Datové oblasti pro potřeby hlášení chyb a identifikaci řádků pro potřeby oprav již zaslaných Hodnot údajů. Datová oblast (dynamická) může obsahovat maximálně jeden identifikační parametr. Oborem hodnot identifikačního parametru je Datový typ,
    - **parametry nesloužící k popisu struktury výkazu** (zde jsou uvedeny jen jako doplňující informace):
        - **výskytový:** parametr, který je použit při definování Vykazovacích povinností. V rámci výskytového parametru je vymezen rozsah Vykazující osoby a stav ke dni,
        - **organizační:** parametr, který je použit pro definování Vykazovacích povinností, stanovuje např. termín předkládání,
        - **technologický:** speciální typ parametru, který slouží pro administraci systému,
b) z hlediska použití ve struktuře Datové oblasti
    - **dimenzionální:** parametr je umístěn na kterékoliv ose nebo osách Datové oblasti a tvoří jednu z dimenzí v její struktuře,
    - **nedimenzionální:** parametr je použit pro dokonkretizaci jednotlivých údajů v Datové oblasti.

3.17.1 Atributy objektu Parametr

Objekt Parametr obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů).

Nad rámec standardních atributů jsou u objektu Parametr definovány následující atributy:

- **typ parametru (number; číselníková položka):** určuje typ použitého parametru (viz výše). Tento typ je dán statickým číselníkem bez možnosti změnit jeho obsah uživatelem,
- **vazba na konkrétní Číselník nebo konkrétní Datový typ:** určituje základní vymezení očekávané hodnoty zaslaného údaje. V objektovém modelu reprezentován vazbou objektu Parametr na objekt Datový typ a Číselník XOR: právě jeden z nich, povinně.

Asociační třída „Konkretizovaný parametr“ pak má následující atributy:

- **pořadí (number):** obsahuje informaci o tom, v jakém pořadí má být daný Konkretizovaný parametr v rámci osy, na kterou je umístěn, prezentován. Každý Konkretizovaný parametr smí mít právě jedno pořadí a toto pořadí musí být jedinečné (nesmí existovat dva Konkretizované parametry v jedné Datové oblasti na jedné ose, které by měly stejné pořadí),
- **osa (number; číselníková položka, nemá default):** definuje, na jaké ose bude daný Konkretizovaný parametr prezentován. Číselník obsahuje tyto možné hodnoty:
  - horizontální osa (osa X),
  - vertikální osa (osa Y),
  - karta (osa Z).

Kombinace Pořadí/Osa musí být v rámci Datové oblasti jedinečné.

3.18 Objekt Údaj

Sledování historie objektu: Bez sledování historie (viz kapitola 2.2.3 Přístup „Bez sledování historie“)

Z hlediska celého systému zcela zásadní objekt, v rámci kterého se propojí veškerá nastavení metapopisu (knihovny). Účelem tohoto objektu je zcela detailně popsat konkrétní „buňku“ Výkazu tak, aby byla popsána vyčerpávajícím způsobem, kterému bude rozumět Vykazující osoba natolik, že zcela přesně bude schopna dodat požadovaný údaj. Požadovaným údajem může být číslo, text, položka číselníku; v podstatě cokoli, co uživatel pomocí metapopisu dokáže nadefinovat. Požadovaným údajem může být v některých případech i binární soubor (nejčastěji PDF dokument nebo sešit aplikace MS Excel).

Objekt Údaj v systému může vznikat automatizovaně (generuje ho systém na základě nastavení metapopisu) nebo jej může vytvořit uživatel (manuálně vznik Údaje je předpokládán v případě projektování výkazů, které v dnešní době obsluhuje aplikace SIPRES), v každém případě pro objekt Údaj platí následující:

- Údaj vzniká vždy v rámci právě jedné Datové oblasti. Bez existence Datové oblasti nemůže Údaj vzniknout. V případě smazání Datové oblasti musí dojít ke smazání všech Údajů, které v rámci Datové oblasti vznikly,

- Údaj v rámci Datové oblasti vzniká jako kombinace právě jedné instance třídy Ukazatel a jedné až N (neomezeně) instance objektu Vykazovaný parametr (viz kapitola 3.18.1 Vznik Údaje – statická Datová oblast),
- může nastat situace, kdy v jiné Datové oblasti vznikne identický Údaj jako již existuje (je popsán stejným Ukazatelem a stejnou sadou Vykazovaných parametrů). Pokud taková situace nastane, systém toto rozpozná během procesu generování Údajů (viz kapitola 7.19 Údaj a funkční požadavek UDJ_1.0), údaj vygeneruje (v systému tak bude existovat dva a více stejně popsaných Údajů), nicméně u takového Údaje nastaví atribut „vykazovat“ na hodnotu „ne“ a takový Údaj propojí na již existující stejný Údaj. Za tím účelem je v systému definována vazba objektu Údaj sama na sebe s názvem „Zobrazovaný údaj“. Tato vazba je definována jako nepovinná (0..1), tzn., že instance objektu Údaj nemusí udržovat odkaz na jinou instanci objektu Údaj, ale pokud ji udržuje, tak platí, že jeden Údaj se smí odkazovat na právě jeden zobrazovaný údaj,
- z hlediska definice byznys pravidel platí, že Údaj je vykazován v právě jedné Datové oblasti a tato Datová oblast je určena uživatelem. Vykazující osoby si nemohou volit, v rámci jaké Datové oblasti budou Údaj vykazovat, musejí se řídit předpisem, který definuje uživatel v rámci metapopisu.

Pro vymezení polohy údaje v Datové oblasti jsou definovány:

- **souřadnice údaje**: je pozice Údaje v Datové oblasti a je vymezena pozicí v rámci sloupců (osa X – **souřadnice sloupce**), řádků (osa Y – **souřadnice řádku**) a karet (osa Z – **souřadnice karty**) jedné Datové oblasti.

3.18.1 Vznik Údaje – statická Datová oblast

Pro vznik Údaje v rámci statické Datové oblasti platí následující pravidla:

- Údaj je připojen k Datové oblasti vazbou typu „kompozice“. To znamená, že Údaj je nedílnou součástí Datové oblasti, Datová oblast musí existovat předtím, než vznikne Údaj, Údaj nemůže vzniknout, aniž by byl zařazen do právě jedné Datové oblasti,
- aby mohl Údaj vzniknout, je nejdříve třeba, aby do patřičné Datové oblasti byly zařazeny Ukazatele a Parametry (tzv. společné parametry), které Údaj popisují,
- do jedné Datové oblasti je možno přiřadit N (neomezeně) Ukazatelů a N (neomezeně) Parametrů,
- jeden Ukazatel smí být do Datové oblasti přiřazen N-krát, pokud však bude přidán více než jednou, musí být zajištěno, že bude pokaždé jinak konkretizován. Konkretizace Ukazatele v Datové oblasti musí být jedinečná (podrobněji jsou tato pravidla popsána v kapitole 3.16.2 Objekt Ukazatel v Datové oblasti),
- jeden Parametr smí být do Datové oblasti přiřazen maximálně jednou. Pokud v Datové oblasti uživatel vytvoří části osy (tj. rozčlenění jednotlivých os do více homogenních částí), pak vložení Parametru do osy či její části vede na přidání Parametru do všech částí dané osy dané Datového oblasti. Parametr se tak vyskytuje vždy ve všech částech dané osy a v každé části osy musí být konkretizovaný (tzv. konkretizace Parametru), může však být v každé části konkretizován jinak - např. Parametr P01 v 1. části osy X s konkretizací na položku (součet), stejný parametr P01 v 2. části osy X s konkretizací na doménu číselníku (detail),

- Parametr se do Datové oblasti přiřazuje přes objekt **Konkretizovaný parametr**. Postup je následující:
    - uživatel určí Datovou oblast, do které chce zařadit Parametr,
    - uživatel určí (výběrem z číselníku) Parametr, který chce do Datové oblasti zařadit. Systém nabídne pouze ty Parametry, které ještě nejsou do dané Datové oblasti zařazeny.
    - během procesu zařazení Parametru do Datové oblasti systém vyžádá od uživatele dodatečnou informaci nutnou k tomu, aby zařazení Parametru do Datové oblasti mohlo být uloženo, tzv. konkretizaci Parametru. Tato konkretizace spočívá v přiřazení právě jedné z následujících informací ke kombinaci Parametr/Datová oblast/část osy:
        - Datový typ,
        - Položka (číselníku nebo hierarchie číselníku),
        - Doména (datového typu nebo číselníku),
        - Hierarchie číselníku,
    - přiřazením této dodatečné informace vzniká instance objektu Konkretizace Parametru. Oproti Ukazateli je situace u Konkretizace parametru složitější o to, že objekt Parametr udržuje již od okamžiku svého vzniku vazbu buď na objekt Číselník anebo na objekt Datový typ. Pokud tedy zařazujeme Parametr do Datové oblasti, musíme tuto vazbu při tvorbě konkretizace respektovat. Pokud zařazujeme Parametr, který je přivázán na Číselník, je možné takový Parametr konkretizovat pouze vazbou na Položku číselníku nebo na Hierarchii číselníku nebo Doménu číselníku (navíc pouze na Položku číselníku/Doménu číselníku/Hierarchii číselníku, která je součástí Číselníku, který je k Parametru připojen). Stejně tak, pokud konkretizujeme Parametr, který je navázán v okamžiku svého vzniku na objekt Datový typ, není možné takový Parametr konkretizovat ničím jiným než touto instancí objektu Datový typ nebo nad ním vytvořenou Doménou datového typu,
    - pokud je v rámci procesu konkretizace Parametru přiřazena na jedné části osy například Doména číselníku, není již možné danému Parametru v rámci dané Datové oblasti v rámci stejné části osy přiřadit jinou konkretizaci (tedy Položku číselníku/hierarchie nebo Doménu datového typu/číselníku nebo Hierarchii číselníku). Kombinace Parametru, Datové oblasti a části osy je jedinečná a obsahuje právě jednu konkretizaci,
- proces vzniku Údaje je zajištěn systémem, postup je následující:
    - před samotným procesem vytváření Údajů systém musí zajistit vytvoření instancí objektu Vykazovaný parametr. Tento objekt zastřešuje všechny jedinečné kombinace Parametr/Položka (číselníku/hierarchie), které lze vytvořit z objektu Konkretizovaný parametr. Proces vzniku instancí objektu Vykazovaný parametr lze demonstrovat na následujícím příkladu:
        - existuje parametr „Měna“, který je konkretizován v rámci Datové oblasti „DO1“ pomocí domény „DOM1“ postavené nad číselníkem „Měny“. Doména „DOM1“ obsahuje tyto položky hierarchie:
            - Všechny měny (součet za všechny položky dané domény hierarchie),
            - EUR,
            - CZK,

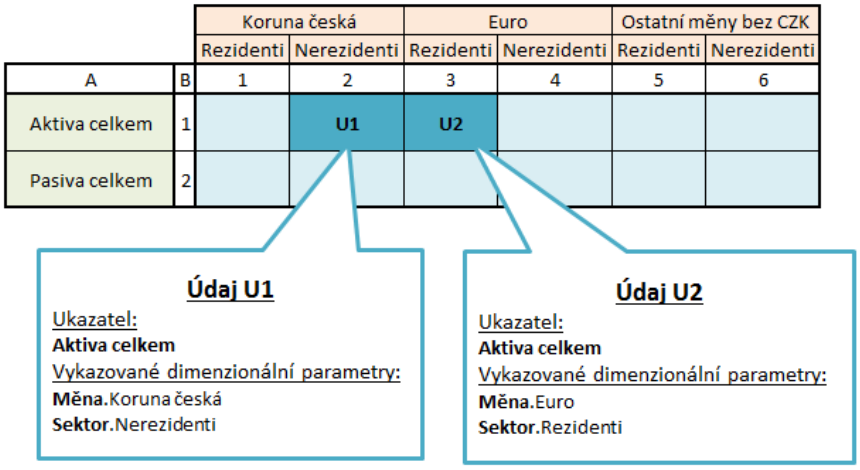
    - Ostatní bez CZK/EUR,
- výše uvedené je zajištěno jedinou instancí třídy Konkretizace parametru KP1 (první instance této třídy v rámci Datové oblasti „DO1“),
- dále existuje parametr „Sektor“, který je konkretizován v rámci Datové oblasti „DO1“ pomocí domény „DOM2“ postavené nad číselníkem „Sektory“. Doména „DOM2“ obsahuje tyto položky hierarchie:
    - Rezidenti/nerezidenti celkem (součet za všechny položky dané domény hierarchie),
    - Rezidenti,
    - Nerezidenti,
- výše uvedené je zajištěno jedinou instancí třídy Konkretizace parametru „KP2“ (druhá instance této třídy v rámci Datové oblasti „DO1“),
- aby bylo možné popsat Údaj, je potřeba:
    - zjistit všechny jedinečné kombinace Parametr/Položka vyplývající ze všech instancí třídy Konkretizovaný parametr v Datové oblasti „DO1“ a vytvořit z nich instance objektu „Vykazovaný parametr“ (objekt „Vykazovaný parametr“ nabývá spíše technického, než věcného významu – je v něm uložen „rozpad“ číselníkových domén na jednotlivé položky číselníku, se kterými se pak snáze pracuje při popisu vytváření údaje)
    - pomocí metody kartézského součinu vynásobit všechny prvky (Položky) všech množin (Parametrů) dané Datové oblasti, tedy vlastně provést kartézský součin všech instancí objektu Vykazovaný parametr a připojit jednotlivé prvky výsledného kartézského součinu Údaji,
    - z výše uvedeného příkladu plyne, že dvě existující Konkretizace parametru „KP1“ a „KP2“ generují celkem 5 instancí třídy Vykazovaný parametr (pro jednoduchost jsou vynechány součtové položky, systém standardně generuje veškeré kombinace, vč. součtových položek):
    - Parametr „Měna“/Položka „CZK“,
    - Parametr „Měna“/Položka „EUR“,
    - Parametr „Měna“/Položka „Ostatní bez CZK/EUR“,
    - Parametr „Sektor“/Položka „Rezidenti“,
    - Parametr „Sektor“/Položka „Nerezidenti“,
- nechť Parametr je množina a Položka je prvek množiny. Máme tak dvě množiny, první o třech prvcích a druhou o dvou prvcích. Metodou kartézského součinu dospějeme k šesti jedinečným kombinacím, tedy:
    - Parametr „Měna“/Položka „CZK“ + Parametr „Sektor“/Položka „Rezidenti“,
    - Parametr „Měna“/Položka „CZK“ + Parametr „Sektor“/Položka „Nerezidenti“,
    - Parametr „Měna“/Položka „EUR“ + Parametr „Sektor“/Položka „Rezidenti“,

- Parametr „Měna“/Položka „EUR“ + Parametr „Sektor“/Položka „Nerezidenti“,
- Parametr „Měna“/Položka „Ostatní bez CZK/EUR“ + Parametr „Sektor“/Položka „Rezidenti“,
- Parametr „Měna“/Položka „Ostatní bez CZK/EUR“ + Parametr „Sektor“/Položka „Nerezidenti“,

o každý Údaj je tak popsán právě jedním Ukazatelem a 1..N (neomezeně, ale alespoň jedním) Vykazovaným parametrem, přičemž N musí u každého Údaje nabývat takového počtu, kolik různých parametrů je Datové oblasti přiřazeno (v uvedeném případě jsou to dva parametry, tudíž jeden Údaj musí být popsán právě jedním Ukazatelem a právě dvěma Vykazovanými parametry (viz Obrázek 13 - Ukazatel a Vykazovaný parametr),

o celkový počet Údajů ve sledované Datové oblasti je tak 12 (stále zanedbáváme vliv součtových položek hierarchie), protože máme 2 Ukazatele a každý z nich vytváří celkem 6 Údajů.

Po rozgenerování všech jedinečných kombinací Parametr/Položka vyplývající ze všech instancí třídy Konkretizovaný parametr v Datové oblasti je možné do defaultní vizualizace metapopisu vybrat/definovat pouze ty kombinace, které jsou skutečně požadovány. Nepožadované kombinace se v defaultní vizualizaci metapopisu vůbec neobjeví.



Obrázek 13 - Ukazatel a Vykazovaný parametr

Pro situaci, kdy jsou jednotlivé osy Datové oblasti členěny do částí osy (viz výše v textu) **neplatí metoda kartézského součinu.** Mezi částmi osy nesmí být při tvorbě instancí objektu Vykazovaný parametr použita tato metoda. Jednotlivé části osy jsou samostatné a jsou zobrazeny jednotlivě za sebou. Kartézský součin je aplikován pouze „uvnitř“ v jednotlivých částech osy.

Výše uvedený algoritmus předpokládá, že v rámci Datové oblasti „DO1“ jsou přiřazeny dva Ukazatele, které jsou konkretizovány Datovým typem. Dále jsou v „DO1“ přiřazeny dva Parametry, které jsou konkretizovány Doménou číselníku nebo Hierarchií číselníku. V takové situaci pak mluvíme o tzv. **dimenzionálních Parametrech.** Tím, že je Parametr konkretizován napojením na Doménu číselníku nebo na Hierarchie číselníku, je řečeno, že tento Parametr může nabývat více než jednu hodnotu. Pokud nabývá více jak jednu hodnotu,

mluvíme o dimenzi. Dále jsou tyto **Parametry společné,** tj. jsou připojeny přímo k Datové oblasti a tím je zajištěno, že jakýkoliv Ukazatel, který bude připojen ke stejné Datové oblasti, bude vždy popsán kartézským součinem všech prvků (Položek číselníku) všech množin (Parametrů).

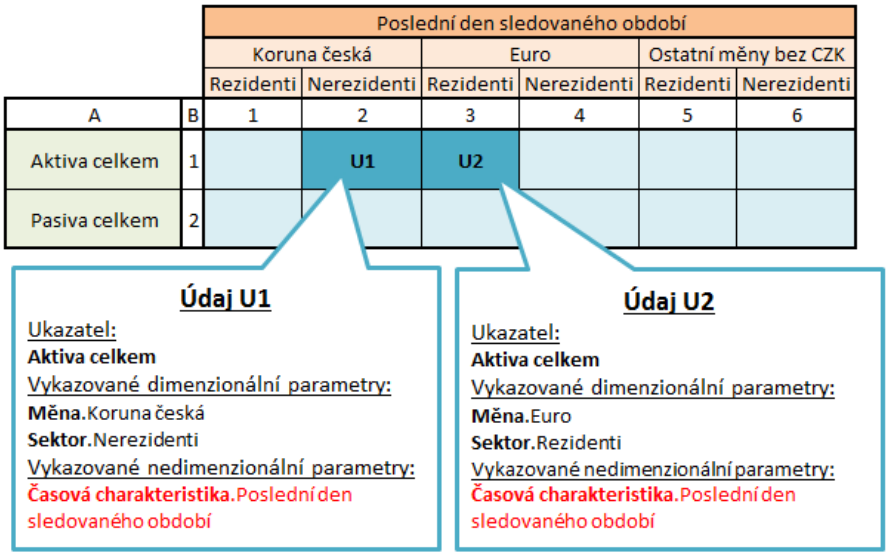
V praxi toto k popisu Údajů nestačí. Je třeba vyřešit následující požadavky:

- popsat Údaj tzv. **nedimenzionálním Parametrem**, tedy parametrem, který je konkretizován tak, že nevznikne dimenze. Nedimenzionální Parametr vzniká tak, že je do Datové oblasti přiřazen Parametr, který je konkretizován právě jednou Položkou číselníku,
- uživatel však může rozhodnout, zda tento nedimenzionální parametr chce vidět v defaultní vizualizaci metapopisu. Pokud ne, parametr a jeho konkretizace (zde jedna položka) zůstane skryta. Pokud se uživatel rozhodne pro vizualizaci, může nastavit zobrazovat textový popisek, tzn. patro (viz Obrázek 14 - Ukazatel a Vykazovaný parametr I.). Implicitně bude nastaveno nezobrazovat.
- pro libovolný Ukazatel zařazený do Datové oblasti zajistit, aby vznikl tzv. **nespolečný popis Údaje**, tedy Údaj, který je popsán nějakým dalším Konkretizovaným parametrem, který není společný pro všechny další Ukazatele zařazené do Datové oblasti.

3.18.2 Popis Údaje nedimenzionálním parametrem

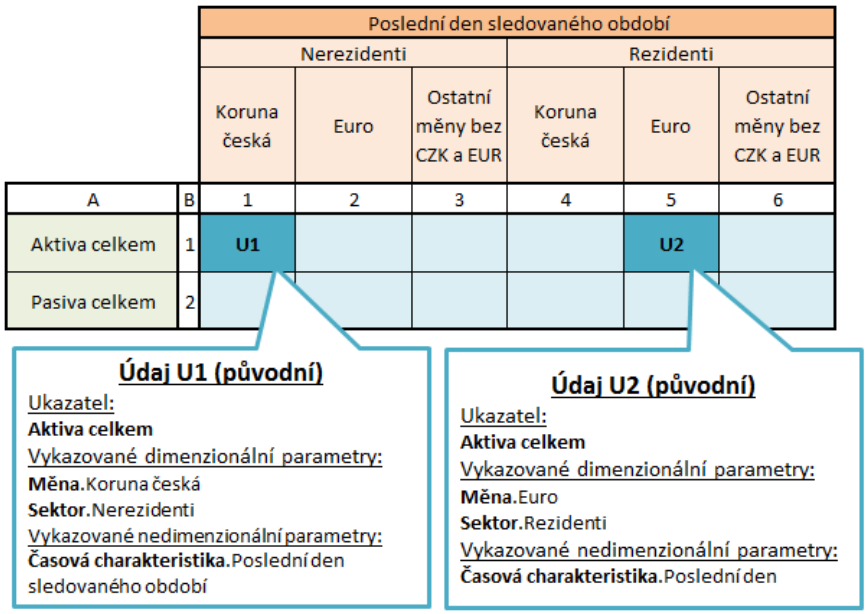
Z hlediska výše uvedeného algoritmu je přidání nedimenzionálního parametru velmi jednoduchá záležitost:

- do Datové oblasti bude zařazen další Parametr, například „Časová charakteristika“ (vůbec neřešíme, jestli Parametr bude či nebude dimenzionálním, ten se dimenzionálním buď stane, nebo nestane, v závislosti na způsobu jeho konkretizace), který na rozdíl od dvou předcházejících Parametrů nebude konkretizován Doménou číselníku postavenou nad Hierarchií číselníku, ale přímo Položkou číselníku („Poslední den sledovaného období“),
- tím, že pro daný Parametr je možná pouze jedna hodnota, tak Parametr nevytvoří žádnou další dimenzi a stane se nedimenzionálním. Kartézský součin sice bude probíhat nad třemi množinami (místo dvou), ale výsledek bude stále stejný 3x2x1=6 (třetí množina, reprezentovaná parametrem „Časová charakteristika“ totiž obsahuje jen jeden prvek – číselníkovou položku „Poslední den sledovaného období“),
- ve výsledku tedy vznikne 6 údajů (pro jeden Ukazatel), stejně jako v předcházejícím případě, rozdíl bude pouze v tom, že nyní každý Údaj bude popsán jedním dalším parametrem (viz Obrázek 14 - Ukazatel a Vykazovaný parametr I.).



Obrázek 14 - Ukazatel a Vykazovaný parametr I.

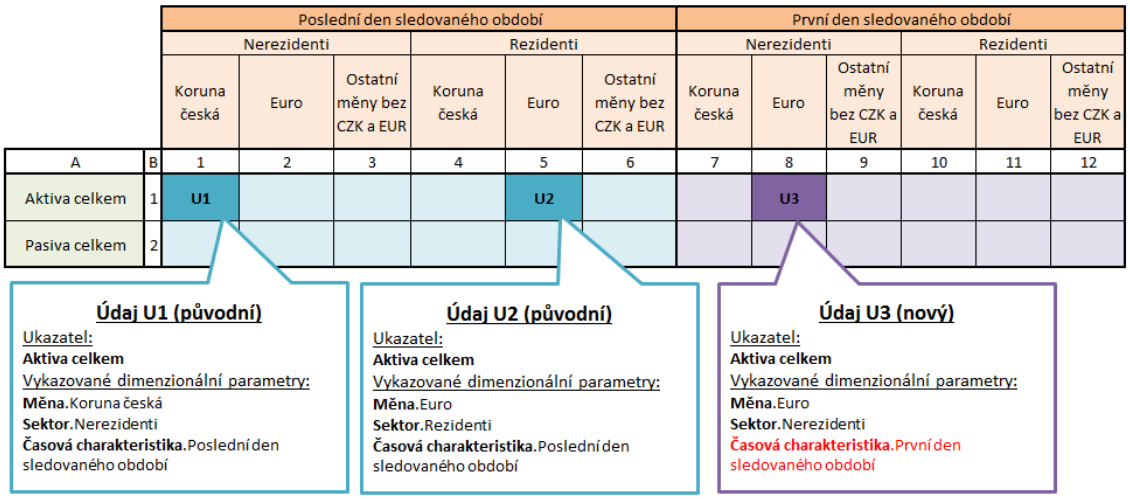
Pokud platí, že Údaj je popsán právě jedním Ukazatelem a 1 – N Vykazovanými parametry, pak můžeme vykazované hodnoty prezentovat libovolným způsobem. Následující obrázek demonstruje situaci, kdy změním v Datové oblasti pořadí Parametrů P1 a P2. Buňky Datové oblasti, které jsou na obrázku vedle sebe, budou po přehození pořadí Parametrů na jiných místech, nicméně jsou stále stejně popsané a udržují odkaz na stále stejné hodnoty.



Obrázek 15 - Ukazatel a vykazovaný parametr II.

V případě, že vznikne potřeba z nedimenzionálního Parametru P3 vytvořit Parametr dimenzionální (v tom případě se změní konkretizace parametru P3; ten nadále nebude konkretizován napojením na právě jednu Položku číselníku, ale bude konkretizován napojením na Doménu číselníku „DOM3“, která bude obsahovat položky „Poslední den

sledovaného období“ a „První den sledovaného období“, dojde k nárůstu definovaných Údajů. V tomto případě se změní konkretizace Parametru P3; ten nadále nebude konkretizován napojením na právě jednu Položku číselníku, ale bude konkretizován napojením na Doménu číselníku „DOM3“, která bude obsahovat položky „Poslední den sledovaného období“ a „První den sledovaného období“. Protože platí pravidlo o kartézském součinu, pak přestane platit vzorec 3x2x1=6, ale tím, že se místo jedné Položky číselníku Parametr P3 konkretizoval pomocí Domény číselníku, která obsahuje 2 položky číselníku, vznikne vzorec 3x2x2=12. Ve výsledku tedy k jednomu Ukazateli vznikne 12 Údajů, celkově jich Datová oblast má 24 (12 Údajů x 2 Ukazatele). Protože nová konkretizace Parametru P3 obsahuje i položku „Poslední den sledovaného období“ (tato položka byla obsažena i v původní konkretizaci), tak se popis původních Údajů nezmění, pouze přibydou Údaje nové (na obrázku 15 zvýrazněné fialovou barvou).



Obrázek 16 - Ukazatel a Vykazovaný parametr III – nedimenzionální parametr se stává dimenzionálním.

3.18.3 Atributy objektu Údaj

Objekt Údaj obsahuje následující standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů):

Interní identifikátor objektu, Poznámka, Autor objektu, Datum a čas vytvoření.

Nad rámec standardních atributů jsou u objektu Údaj definovány následující atributy:

- **stupeň citlivosti údaje (number; číselníková položka):** definuje, jak je údaj citlivý z hlediska zpracování jeho hodnoty. Stupeň citlivosti dat může nabývat právě jedné hodnoty z níže uvedeného seznamu hodnot:
  - necitlivý údaj (nejnižší stupeň citlivosti),
  - citlivý údaj.

  Citlivost je sledována na každém jednom Údaji, a pokud uživatel neřekne jinak, má se za to, že Údaj (resp. jeho hodnota) je necitlivý a nevyžaduje přidělení žádného speciálního oprávnění jako je tomu v případě citlivých údajů, a podléhá tak standardním oprávněním.

  Informace o citlivosti jsou propagovány z Údaje na nadřízené objekty. Je tedy požadováno, aby existovala „citlivost Datové oblasti“ a „citlivost Výkazu“. Tyto

nadřízené objekty ale nemají svůj vlastní atribut pro citlivost, hodnota citlivosti těchto objektů je odvozena z citlivosti Údajů v nich obsažených, tedy Výkaz je citlivý pouze v případě, že obsahuje alespoň jeden Údaj, jehož stupeň citlivosti nabývá hodnotu „citlivý údaj“. Pokud má Výkaz Datovou oblast, kde je deset Údajů, z čehož je 9 označeno stupněm citlivosti „necitlivý údaj“ a 1 jako „citlivý údaj“, pak stupeň citlivosti celého Výkazu je „citlivý“,

- **vykazovat (boolean; default je „ano“):** definuje, zda má být daný Údaj předmětem vykazování. Standardně je nastaveno, že každý vygenerovaný Údaj je nastaven jako „vykazovaný“, nicméně uživatel může libovolný Údaj tzv. „vykřížkovat“, tedy určit, že daný Údaj nemá být předmětem vykazování ze strany Vykazujících osob. Takový Údaj je pak prezentován jako „neaktivní“. Podrobně je účel tohoto atributu uveden ve funkčním požadavku UDJ_2.0,
- **zobrazovaný údaj:** jedná se o tzv. odvozený atribut, který vznikne z nepovinné rekurzivní asociační vazby „zobrazovaný údaj“ který uchovává informaci o nadřízeném Údaji. Podrobně je účel tohoto atributu uveden ve funkčním požadavku UDJ_1.0,
- **popisek údaje:** možné využít pouze pro typ Datové oblasti – volná. Definuje popisek, který bude dále možné využít jako „label“ formulářového pole při projektování specifických výkazů (výkazy projektované v současné době v aplikaci SIPRES).

3.18.4 Dynamické atributy Údaje

Systém umožní ke každému Údaji specifikovat dodatečné atributy. Příkladem mohou být atributy přesnost nebo měna (odpovídají konstruktům *decimals* a *unitRef* u výkazů převzatých z XBRL taxonomií) nebo citlivost hodnoty údaje (*confidentiality*). Dynamické atributy údajů definuje projektant. Kromě jejich názvu, datového typu a povinnosti stanovuje jejich výchozí hodnotu (např. v případě XBRL atributu decimals je to maximální přípustná hodnota, kterou stanovuje metodický předpis ITS). Systém umožňuje provádět tuto definici a nastavení výchozí hodnoty hromadně pro skupinu Údajů, tj. na úrovni Výkazu nebo Datové oblasti pro všechny jeho Údaje ve vazbě na použité datatypy (základní datové typy) (filtrem je výkaz a základní datový typ – datatyp – údaje). Zároveň systém umožnuje provést tuto operaci i pro jeden konkrétní Údaj.

Takto definované atributy údajů jsou pak předmětem vykazování, tj. jejich hodnota je vykazována společně s Hodnotou údaje. Hodnoty dynamických atributů Údajů jsou zasílány Vykazující osobou v rámci Vydání výskytu výkazu jako součást Hodnoty údaje[2] a jsou společně ukládány. Rovněž mohou být použity v rámci procesu zpracování Vydaní výskytu výkazu (např. v kontrolách). Pokud Vykazující osoba zašle Vydání výskytu výkazu bez hodnot povinných dynamických atributů údaje, je takové Vydání výskytu výkazu na základě kontrol odmítnuto.

V rámci webové aplikace pro vykazování systém obsahuje funkci, která umožní Vykazující osobě vložit hodnotu atributu a to buď ke každé Hodnotě údaje zvlášť, nebo hromadně na úrovni Výkazu nebo Datové oblasti ve vazbě na datový typ. Zároveň systém umožnuje použít výchozí hodnotu nastavenou projektantem výkazu.

[2] může se jednat např. o atributy nebo dodatečné elementy, pokud by formátem přenosové zprávy bylo XML

3.19 Objekt Kontrola

Sledování historie objektu: N/A (historie se sleduje u potomků).

Systém SDAT rozlišuje dva typy kontrol:

- věcné kontroly
- technické kontroly.

Věcná kontrola specifikuje vztahy mezi Údaji, které musí být naplněny, aby daná Hodnota údaje byla prohlášena za správnou. Věcné kontroly jsou vykonávány systémem SDAT vždy při vstupním zpracování dat Výkazu v ČNB. Zároveň je umožněno, aby stejný rozsah kontrol (s výjimkou algoritmických kontrol) si mohla provést Vykazující osoba ve svém vlastním prostředí před odesláním dat do ČNB prostřednictvím internetové aplikace nebo externím interpretem kontrol SDAT (off-line interpret kontrol, který umožňuje vykonat kontroly v prostředí Osoby, tedy bez nutnosti předávat data do ČNB) nebo použitím jiného programového řešení, které umí kontroly interpretovat na základě metapopisu.

Technické kontroly představují kontroly vyplývající z vlastností objektů použitých pro definici konkrétního Výkazu a také z technického řešení přenosu dat. Zaslaná data musí splňovat všechny technické náležitosti podle zvoleného přenosového formátu (syntaktické a formální kontroly Vstupní zprávy, kontroly těla Vstupní zprávy). Technické kontroly jsou systémem prováděny při vstupním zpracování a v případě jejich nesplnění nejsou uložena data do databáze (nevzniká instance objektu Hodnota údaje). Součástí technických kontrol je kontrola formátu zaslaných jednotlivých Hodnot údajů, které jsou uživatelem definovány při tvorbě výkazu (formátové kontroly Hodnot údajů). Problematika rozsahu a provádění technických kontrol je popsána v dokumentu D – Sběr dat, kapitola 3.3 Proces zpracování vstupní zprávy. Technické kontroly nejsou předmětem objektu Kontrola (a jeho potomků).

Věcné kontroly systému SDAT se dále člení na:

- jednovýkazové kontroly (JVK),
- mezivýkazové kontroly (MVK),
- kontroly časových řad (KČŘ),
    - mezisubjektové kontroly (MSK).

V objektovém modelu je zaveden objekt Kontrola, ke které jsou vazbou dědičnosti připojeny abstraktní třídy Jednoduchá kontrola a Komplexní kontrola. Tyto třídy byly zavedeny proto, aby mohly být z pohledu dědičnosti vlastností postaveny na jednu úroveň všechny typy kontrol bez ohledu na to, zda se váží k jednomu nebo více Výkazů. Objekt Jednoduchá kontrola představuje předka pro typy kontrol, které vážou vždy k právě jednomu Výkazu (JVK, KČŘ a MSK). Objekt MSK je přímým potomkem objektu KČŘ. Objekt Komplexní kontrola je předkem kontrol, které vážou na více než jeden Výkaz (MVK). Samotné napojení objektu MVK na objekt Výkaz je realizováno prostřednictvím objektů Skupina MVK a Člen MVK, tedy jinak, než je realizováno napojení JVK/KČR/MSK. To je i důvod pro zavedení obou výše zmíněných abstraktních tříd.

Objekt Skupina MVK sdružuje jednotlivé MVK přes zapojené Výkazy za účelem jejich lepší spravovatelnosti. K objektu Výkaz je objekt MVK připojen přes asociační třídu Člen MVK, která umožňuje definovat jaké Výkazy a v jakém postavení (člen/vlastník) jsou Výkazy do konkrétní Skupiny MVK zařazeny. Jeden Výkaz může mít N (neomezeně; tedy i žádnou) kontrol. V případě ukončení platnosti Výkazu jsou ukončeny všechny kontroly (JVK,

KČŘ/MSK, MVK), které jsou napojeny k danému výkazu. V případě smazání Výkazu ve stavu Projektovaný jsou smazány všechny verze kontrol, které vznikly v rámci mazané verze Výkazu.

3.19.1 Atributy objektu Kontrola

Objekt Kontrola obsahuje všechny standardní atributy (viz kapitola 2.4.2 Standardní atributy objektů) s výjimkou následujících:

- Garant.

Pro následující standardní atributy objektu Kontrola platí tato specifika:

- **kód objektu:** je jednoznačný v určitém časovém řezu v rámci Kontrol patřících k jednomu Výkazu,
- **název objektu:** v případě ručních kontrol přiřazuje uživatel, v případě kontrol generovaných systémem přiřazuje systém na základě názvové konvence s možností název uživatelsky změnit.

Nad rámec standardních atributů jsou u objektu Kontrola definovány následující atributy:

- **úroveň závažnosti:** atribut nabývá následujících hodnot:
  - **závažná chyba**: při nesplnění této Kontroly alespoň u jedné Hodnoty údaje Vydání výskytu výkazu, není splněna Vykazovací povinnost a od Vykazující osoby se očekává zaslání Vydání výskytu výkazu typu Oprava/změnová oprava. Výsledek kontroly se promítá do kvality Hodnoty údaje,
  - **chyba k potvrzení**: při nesplnění této kontroly alespoň u jedné Hodnoty údaje Vydání výskytu výkazu není splněna Vykazovací povinnost a od Vykazující osoby se očekává zaslání následného Vydání výskytu výkazu typu Potvrzení nebo Oprava. Výsledek kontroly se promítá do kvality Hodnoty údaje,
  - **varování**: nesplnění této kontroly nemá vliv na plnění Vykazovací povinnosti ani na kvalitu Hodnoty údaje,
- **sémantický tvar vzorce kontroly:** Údaje vstupující do vzorce kontroly jsou zachyceny pomocí parametrického popisu,
- **uživatelský tvar vzorce kontroly:** Údaje vstupující do kontroly jsou zachyceny pomocí jejich souřadnic ve struktuře Výkazu (řádek, sloupec, případně karta).

  V případě, že se jedná o algoritmickou kontrolu, je obsahem atributů sémantický tvar vzorce kontroly a uživatelský tvar vzorce kontroly pouze identifikátor Údaje vstupujícího do kontroly v příslušném tvaru zápisu.

- **validní:** atribut nabývá hodnot ano (validní kontrola), ne (nevalidní kontrola – vzorec kontroly obsahuje chybu, která brání vyhodnocení kontroly),
- **dokončená:** atribut nabývá hodnot ano (dokončená kontrola – vzorec kontroly je z věcného hlediska hotov), ne (rozpracovaná kontrola),
- **automatická:** atribut nabývá hodnot ano (automatická kontrola vytvořená dle pravidel systémem), ne (ruční kontrola vytvořená uživatelem)

3.19.2 Způsob vytvoření kontroly

Z pohledu způsobu vytváření věcných kontrol systém SDAT rozlišuje kontroly vytvořené a zapsané sémantickým jazykem a algoritmické kontroly. Těmito způsoby mohou být popsány všechny typy kontrol, tj. JVK, MVK, KČŘ, MSK.

3.19.2.1 Kontroly vytvořené sémantickým jazykem

Vzorce kontrol jsou zapisovány pomocí předem definovaného jazyka, který umožňuje odkazovat na jednotlivé elementy Výkazů pomocí sémantiky metapopisu a používat sadu funkcí a operátorů určenou k sestavování kontrolních výrazů. Definice (vzorce) těchto kontrol je součástí metapopisu, na základě kterého je lze jednoznačně interpretovat pomocí softwarových prostředků. Tyto kontroly lze provádět off-line i mimo prostředí systému SDAT externím interpretem SDAT nebo vlastním řešením na základě čtení vzorců kontrol z metapopisu.

3.19.2.2 Algoritmické kontroly

Základem kontroly je funkce s parametry, která je před jejím použitím v rámci kontroly vytvořena programátorem. Některé níže popsané funkce pro algoritmické kontroly jsou součástí dodávky systému SDAT. Zároveň systém umožňuje programátorovi z ČNB připravit funkci v podobě jedné procedury nebo i jako součást komplexnějšího balíku procedur a v součinnosti s technickým správcem systému v ČNB ji umístit a uživatelským způsobem zaevidovat do systému. Samotný zápis je realizován pomocí programového kódu minimálně v programovacím jazyku PL/SQL. Do systému umístěnou a evidovanou funkci (nabízí se uživatelům např. ze seznamu) lze následně použít při vytváření kontrol, tj. provést parametrizaci funkce. Algoritmus (vzorec) kontroly však není přímo součástí metapopisu. Pro tento typ kontrol je dostupná pouze informace o její existenci a textový popis algoritmu, jehož vložení do systému je součástí výše zmiňovaného zaevidování funkce.

Algoritmické kontroly mohou volat procedury a využívat datové zdroje umístěné mimo systém SDAT nebo i mimo prostředí ČNB (kontroly vůči externím registrům spravovaných například Českým statistickým úřadem atd.). Ve všech případech se však jedná o synchronní komunikaci, tj. i v případě využití cizího zdroje kontrola čeká na výsledek volání.

Tento typ kontrol lze provádět off-line mimo systém SDAT pouze vlastním programovým řešením algoritmu u Osoby.

Pro použití obou typů kontrol platí v SDAT následující pravidlo. Maximum kontrol je zapisováno sémantickým jazykem. Pouze ty, které vůbec nelze takto vytvořit, nebo ty pro které je sémantický zápis neefektivní (např. z výkonových důvodů) je použito řešení pomocí algoritmické kontroly.

3.19.3 Objekt Jednovýkazová kontrola (JVK)

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Objekt Jednovýkazová kontrola (JVK) představuje vzorec, který kontroluje Hodnotu údaje nebo vzájemný vztah více Hodnot údajů v rámci jednoho Výkazu (Vydání výskytu výkazu) a období (stav ke dni).

JVK jsou definovány buď automatizovaně systémem na základě definovaných hierarchií v Datové oblasti (hierarchie Ukazatelů a hierarchie Položek číselníku v rámci použité Domény číselníku nebo Hierarchie číselníku) nebo ručně uživatelem.

Tyto kontroly lze zapsat jak sémantickým jazykem, tak algoritmicky.

3.19.3.1 Kontroly vytvořené sémantickým jazykem

V závislosti na typu Datové oblasti lze pro definici JVK používat následující operátory, funkce a výrazy:

1) statické Datové oblasti:
   - klasické matematické operace:
     - součet, rozdíl, násobení, podíl (mezi jednotlivými řádky nebo sloupci),
     - absolutní hodnota,
   - srovnávací operátory:
     - základní - rovnost, větší/menší než,
     - skupinové – IN (…), NOT IN (…),
   - množinové funkce:
     - maximum a minimum pro zjištění extrémní hodnoty z definované množiny hodnot,
     - počet, např. pro omezení počtu zaslaných hodnot údajů z definované množiny hodnot),
     - aritmetický průměr z definované množiny hodnot,
     - suma definované množiny číselných hodnot,
   - logické operátory:
     - „a současně“ (AND),
     - „nebo“ (OR),
     - „negace“ (NOT),
   - **konstrukce podmínky**, např. je-li nějaký údaj roven/větší/menší než jiný údaj, pak platí „klasický matematický vztah“),
   - použití konstanty v rámci výrazu,
   - **použití substituce znaků za „wildcards“,** tj. při definování vzorce zapsat i část Hodnoty údaje nebo hodnoty parametru, které má/nesmí dosahovat, např.:
     - např. AI% - na prvních 2 pozicích je AI, zbytek libovolný,
     - poslední 2 znaky v rozmezí nn – NN,
   - regulární výrazy,
   - stanovit tolerovanou **odchylku** Hodnoty údaje pro potřeby kontrol. Další informace k této funkci viz kapitola 3.19.6 Odchylka v sémantických kontrolách,
2) dynamické Datové oblasti:
   - stejné typy kontrol jako u statických Datových oblastí pro jednotlivé řádky dynamické Datové oblasti,
   - kontrola součtu (rovnost nebo větší než) jednotlivých řádků dynamické Datové oblasti (rozdílný počet u jednotlivých Vykazujících osob) na celkový součet,

- omezení provádění definovaných kontrol jen na definované dynamické řádky,
- kontroly mezisoučtů (součtu definovaných) dynamických řádků,
- v rámci definovaného dynamického řádku nepovolit některé kombinace hodnot Parametrů a Ukazatelů,
- kontroly mezi jednotlivými řádky více dynamických Datových oblastí,
- omezení počtu zaslaných dynamických řádků,
- kontroly nad hodnotami dynamických Parametrů a Ukazatelů.

3) kartotékové Datové oblasti:
   - stejné typy jako u statických a dynamických Datových oblastí,
   - omezení počtu zaslaných listů kartotéky,
   - kontroly mezi jednotlivými listy kartotéky.

3.19.3.2 Algoritmické kontroly

Některé JVK kontroly lze řešit pouze algoritmicky, protože jejich realizace je nad rámec schopností sémantického jazyka kontrol. Následující typy jsou příkladem takových kontrol. Konkrétní funkce (programové procedury) pro tvorbu těchto kontrol budou vznikat v rámci provozu systému SDAT:

- **kontrola údaje v externím registru:** typicky se jedná o kontrolu identifikátoru osoby nebo nástroje vůči číselníku/registru mimo systém SDAT. Např. kontrola existence IČO v Registru ekonomických subjektů, cenného papíru v nějaké externí databázi cenných papírů aj.
- **kontrola údaje oproti ISO normě:** typicky se jedná o kontrolu identifikátoru osoby nebo investičního nástroje oproti pravidlům stanoveným ISO normou, tj. tato norma musí být naprogramována v PL SQL a následně použita v SDAT. Např. rodné číslo musí být podle ISDP, ISIN musí odpovídat ISO 6166, CFI musí odpovídat ISO 10962 aj.
- **kontrola údaje v rámci celého výskytu nebo v rámci více výskytů:** kontrolované údaje nebudou jen z jednoho vydání, nýbrž v rámci celého výskytu (jeden stav ke dni) nebo v rámci více výskytů (více stavů ke dni). Využití např. u výkazů Kapitálových trhů.

3.19.4 Objekt Kontrola časové řady (KČŘ)

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Objekt Kontrola časové řady (KČŘ) je vzorec (nebo spíše algoritmus), kterým se kontrolují Hodnoty údaje v rámci více Vydání výskytu výkazu jednoho Výkazu (časová řada hodnot). Počet Vydání výskytu výkazu jednoho Výkazu, jež se účastní kontrol, není předem znám.

Tento typ kontrol je vytvářen pouze algoritmicky. Patří sem kontroly odchýlení hodnoty údaje od lineární regresní přímky a kontroly jedinečnosti údaje, které jsou požadovány jako dodaná součást systému SDAT (podrobná specifikace viz kapitola 8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systém SDAT).

3.19.4.1 Objekt Mezisubjektová kontrola (MSK) jako specifický typ KČŘ

Potomkem objektu KČŘ je objekt **Mezisubjektová kontrola (MSK),** pro kterou platí, že se kontrolují údaje z více Vydání výskytu výkazu vůči údajům z jiných Vydání výskytu výkazu jiných Vykazujících osob. Jedná se tedy o kontroly mezi různými Výkazy (Vydáními výskytu výkazu) a různými Vykazujícími osobami. Oproti Mezivýkazovým kontrolám se liší tím, že počet Vydání výskytu výkazu jednoho Výkazu, jež se účastní kontrol, není předem znám.

Tento typ kontrol je vytvářen pouze algoritmicky a je požadován jako dodaná součást systému SDAT (podrobná specifikace viz kapitola 8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systém SDAT).

3.19.5 Objekt Mezivýkazová kontrola (MVK)

Sledování historie objektu: Sledování historie – časová platnost (viz kapitola 2.2.5 Přístup „Sledování historie – časová platnost“)

Objekt Mezivýkazová kontrola (MVK) je vzorec, který kontroluje údaje v rámci jednoho Výkazu mezi více obdobími nebo mezi více Výkazy k jednomu nebo více obdobím. Podmínkou je, že se jedná vždy o data jedné Vykazující osoby.

Tyto kontroly lze zapsat jak sémantickým jazykem, tak algoritmicky.

Kompletní objektový model objektů souvisejících s prováděním MVK je zachycen v dokumentu D – Sběr dat, kapitola 2.7 Objekt Mezivýkazová kontrola.

3.19.5.1 Kontroly vytvořené sémantickým jazykem

V závislosti na typu Datové oblasti lze pro definici MVK používat následující operátory, funkce a výrazy:

1) statické Datové oblasti:
   - shodné typy jako u Jednovýkazových s tím, že jsou prováděny nad různými Výkazy ke shodnému stavu ke dni a/nebo proti předcházejícím obdobím,
2) dynamické Datové oblasti:
   - shodné typy jako u JVK s tím, že jsou prováděny nad různými Výkazy ke shodnému stavu ke dni a/nebo proti předcházejícím obdobím,
   - součet řádků výkazu v příslušném sloupci (příp. součtu/rozdílu/násobku/podílu sloupců) se rovná konkrétnímu poli v jiném Výkazu/Výkazech (může jít např. o součet údajů ze dvou Výkazů, který se rovná hodnotě ve třetím Výkazu) ve stejném čase,
   - konkrétní údaj v příslušném sloupci v jednotlivých řádcích Výkazu se rovná konkrétnímu údaji ve sloupci v jednotlivých řádcích v jiném Výkazu ve stejném čase (propojení Výkazů mezi sebou, tj. pokud je uvedena nějaká osoba ve výkazu o obchodech, musí se najít ve výkazu o osobách apod.),
   - konkrétní údaj v příslušném sloupci v jednotlivých řádcích Výkazu se rovná konkrétnímu údaji ve sloupci v jednotlivých řádcích v jiném Výkaze za delší časové období např. 6 měsíců (propojení Výkazů mezi sebou, tj. např. nalezení pokynu k obchodu),

- kontrola existence záznamu se shodným identifikátorem v předcházejícím období příp. s určením nějaké podmínky (např. identifikátor pohledávky a kontrola, zda byl v dalším období vykázán, když nemá zároveň vyplněno v jiném poli ukončení pohledávky),

3) kartotékové Datové oblasti:
   - shodné typy jako u JVK s tím, že jsou prováděny nad různými výkazy ke shodnému stavu ke dni a/nebo proti předcházejícím obdobím.

3.19.5.2 Algoritmické kontroly

Jedná se o MVK, které je možné vzhledem k jejich komplexnosti zapsat pouze algoritmem a ten následně aplikovat na Údaj. Typickým příkladem MVK zapsané algoritmem je např. kontrola existence čísla objednávky ve výkazu uskutečněných obchodů oproti číslu objednávky ve výkazu přijatých objednávek, přičemž období mezi příjmem objednávky a jejím uspokojením má proměnlivou délku.

Mezivýkazové algoritmické kontroly, které mají být dodány jako součást systému SDAT jsou uvedeny v 8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systém SDAT.

3.19.6 Odchylka v sémantických kontrolách

Metapopis stanovuje, v jakých násobcích a s jakou přesností Vykazující osoba zasílá jednotlivé Hodnoty údajů (číselné). Toto je dáno Datovým typem, tj. např. v tisících bez desetinného místa, v jednotkách na dvě desetinná místa. Pokud by nebyla splněna kontrola z důvodu zaokrouhlování (ČNB zpravidla předepisuje vykazování v násobcích) musela by Vykazující osoba související Hodnoty údajů upravovat. Systém SDAT proto umožňuje do definice **relačního** výrazu zadat velikost zaokrouhlovací chyby, tj. odchylku, o kterou se Hodnota údaje může odlišovat od požadovaného výsledku. Velikost odchylky se stanovuje v jednotkách (data jsou v databázi uložena v jednotkách) bez ohledu na to, v jakých násobcích se Hodnoty údajů vykazují. Výše odchylky se stanovuje podle počtu operandů (údajů vstupujících do kontroly) následujícím způsobem:

Počet operandů (M)	Odchylka
2	2
3 až 5	3
6 až 19	4
20 až 29	5
30 až 39	6

Tabulka 5 – Odchylky v závislosti na počtu operandů

A dále dle vzorce: 1,0639 * druhá odmocnina (M), kde M je počet operandů.

Příklad 1

Hodnoty údajů U1 až U4 se vykazují v tisících jednotek. Uživatel definuje relační výraz v kontrole včetně odchylky ve výši 3000 (počet operandů 4):

U1 + U2 + U3 = U4 **ODCH** 3000;

Příklad 2

Hodnoty údajů U11 až U31 se vykazují v jednotkách na 2 desetinná místa. Výše odchylky je 0,05:

U11 = U12 + … + U31 **ODCH** 0,05;

Příklad 3

Počet operandů 100, Hodnoty údajů se vykazují v jednotkách, pak se odchylka se stanoví ve výši 11. V případě, že je vykazování Hodnot údajů v milionech, je výše odchylky 11 000 000.

Stanovení odchylky je v relačních výrazech v kontrolách nepovinné. Jedna kontrola může obsahovat 0 až n odchylek (více než 1 odchylku může obsahovat kontrola s více relačními výrazy). Systém na nepřítomnost odchylky nebo nesprávného řádu odchylky upozorňuje uživatele na obrazovce chybovým hlášením. Protože však existují kontroly, které z povahy věci nemohou mít žádnou odchylku, uživateli se při zápisu zobrazuje výzva ke stanovení odchylky v relačním výrazu. Uživatel může výzvu ignorovat, tj. odchylku nezadat.

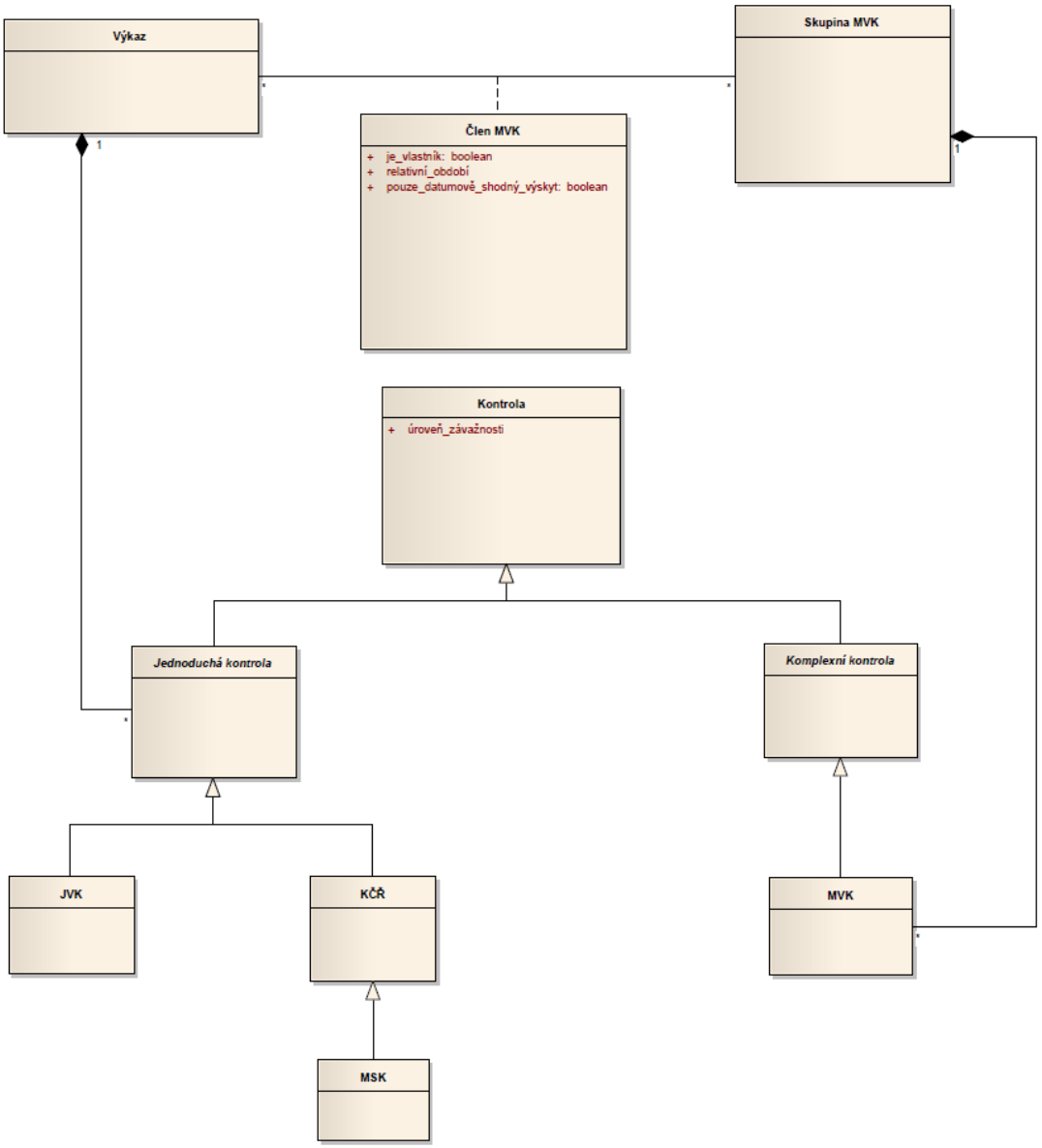
Při generování kontrol z hierarchií systém odchylku stanovuje automaticky pro číselné Hodnoty údajů dle výše uvedeného algoritmu. V případě součtu nad dynamickými Údaji systém generuje odchylku ve výši 10 (+ řád odpovídající násobku vykazovaných hodnot). Uživatel může velikost odchylky upravit podle jím předpokládaného počtu dynamických Údajů ve Vydání výskytu výkazu zasílaném Vykazujícími osobami.

Příklad generované kontroly nad dynamickými údaji

Datová oblast obsahuje součtový řádek U9 a zároveň „neúplný“ dynamický údaj U8, který reprezentuje reálně vykázané Údaje a k nim přiřazené Hodnoty údajů. Hodnoty údajů se vykazují v tisících. Kontrola je generována systémem s výší 10 + řád 1000:

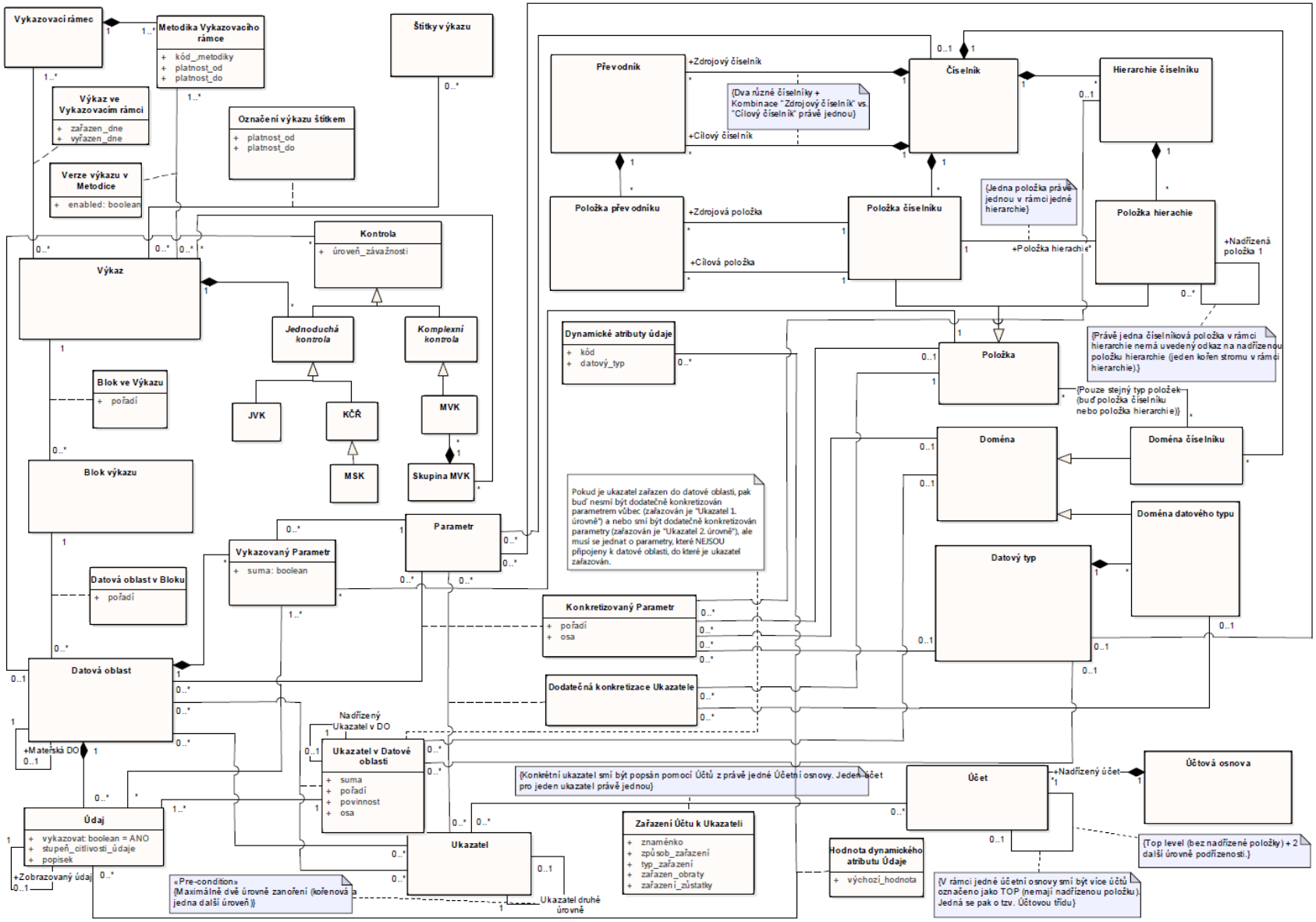
SUM (U8) = U9 **ODCH** 10000.

3.20 Objektový model – Definice kontrol



Obrázek 17 - Objektový model - Kontroly

3.21 Objektový model – Metapopis



Obrázek 18 - Objektový model metapopis

4 Knihovna

Knihovnou se rozumí okruh uživatelského rozhraní, ve kterém jsou shromážděny objekty, které uživatel používá při vytváření popisu Údaje. Knihovna tak uživateli umožňuje přístup k těmto objektům (a aktivitám s těmito objekty souvisejícími):

- Číselník,
- Položka číselníku
- Hierarchie číselníku,
- Položka hierarchie
- Doména číselníku,
- Parametr,
- Datový typ,
- Doména datového typu,
- Ukazatel (1. a 2. úrovně),
- Převodník,
- Položka převodníku,
- Účtová osnova,
- Účet.

Pro práci s objekty Knihovny platí:

- funkce nad objekty Knihovny (založení, modifikace, ukončení platnosti a další aktivity) mohou být prováděny pouze v Knihovně,
- v případě, že uživatel při práci v Metodice vykazovacího rámce narazí na potřebu upravovat objekty, které jsou primárně udržovány v Knihovně, systém umožní uživateli efektivním způsobem přechod do Knihovny, provedení patřičných aktivit a návrat na původní místo, ve kterém prováděl aktivity před tím, než přešel do Knihovny,
- je umožněno vyhledávání objektů Knihovny podle jejich atributů s využitím všech možností tabulky dat (viz GRI_1.0 – GRI_15.0).

Pro ostatní objekty a jejich verze (Výkaz, Blok výkazu, Datová oblast, Kontroly, Vykazovací povinnosti atd.) platí, že vznikají přímo v konkrétní Metodice vykazovacího rámce. Založení, modifikace, ukončení platnosti a další aktivity jsou prováděny zde.

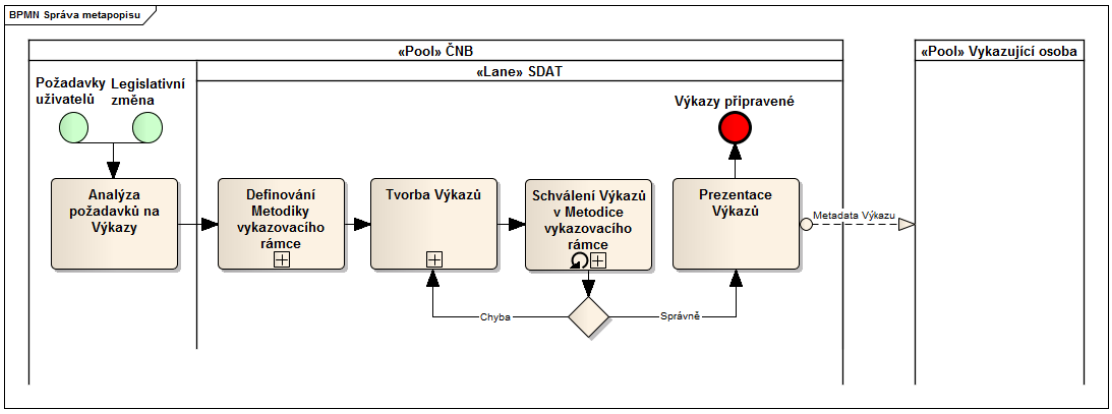
V Metodice vykazovacího rámce je umožněno vyhledávání objektů podle jejich atributů s využitím všech možností tabulky dat.

5 Hlavní procesy

5.1 Proces tvorby metapopisu

Tvorba metapopisu představuje základní aktivitu, jejímž cílem je kompletní popsání požadavků na data metadaty tak, aby byla zajištěna maximální obsahová konzistence mezi jednotlivými Vykazujícími osobami a bylo umožněno následné zpracování dat a jejich použití pro účely analýz, dohledu apod.

Tvorba metapopisu zahrnuje aktivity začínající analýzou požadavků na data, pokračující tvorbou případně úpravou jednotlivých Výkazů a končící prezentací Výkazů Osobám (viz Obrázek 19 - Proces tvorby metapopisu). Jednotlivé části tohoto procesu jsou detailně popsány samostatně v následujících kapitolách. V případě, že některá problematika je již řešena v jiné části, je uveden pouze odkaz na tuto část.



Obrázek 19 - Proces tvorby metapopisu

5.1.1 Spouštěč procesu

Tvorba metapopisu je spouštěna na ad-hoc bázi, tedy podle vzniku potřeby nového Výkazu nebo úpravy již existujícího Výkazu. Zpravidla ke změnám dochází jednou ročně a jsou cíleny k počátku roku, může však být z různých důvodů požadováno i nastavení změn v průběhu roku. Toto rozhodnutí se děje mimo vlastní systém SDAT.

5.1.2 Průběh procesu

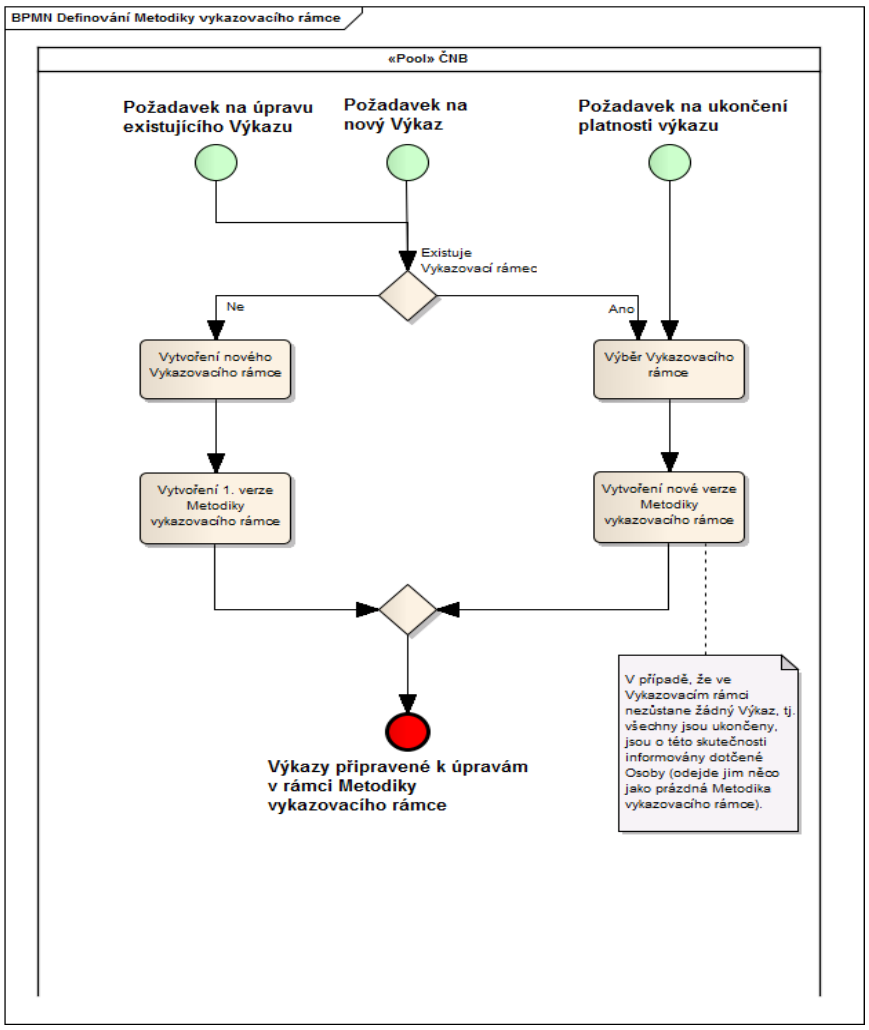
Uživatel vytvoří novou Metodiku vykazovacího rámce, která obsahuje všechny verze výkazů z předchozí metodiky. V Metodice vykazovacího rámce lze změnit nebo jinak upravovat výkazy. Výkazy, které chce uživatel upravovat, je nutné v dané Metodice vykazovacího rámce zaverzovat. Výkazy, které uživatel upravovat nechce, zůstávají ve verzi z předchozí Metodiky vykazovacího rámce. Kromě úprav výkazů lze v Metodice vykazovacího rámce zakládat nové Výkazy (V rámci této Metodiky vykazovacího rámce dochází k tzv. tvorbě výkazu, kdy je tvořen samotný metapopis jednotlivých Výkazů, jejich vizualizace a nastavení věcných kontrol výkazů (viz kapitola 5.3 Proces tvorby Výkazu). Po ukončení tvorby Výkazu, ale před jeho schválením, může uživatel kompletní metapopis vybraného Výkazu předběžně zaslat jím určené skupině Osob k připomínkám nebo testování (viz dokument A – Obecné požadavky, kapitola 2.1.2 Testovací prostředí). Nad vytvořenými Výkazy jsou spouštěny kontroly celkové konzistence. V případě negativního výsledku kontroly celkové konzistence se uživatel vrátí k procesu tvorby výkazu. Po dosažení pozitivního výsledku kontroly konzistence jsou Výkazy schváleny (viz kapitola 5.6 Proces schválení výkazů v Metodice vykazovacího rámce). Schválené Výkazy jsou prezentovány Osobám v různých formách z věcného a technologického hlediska, což jim umožní je implementovat do jejich systémů s předstihem před začátkem platnosti Výkazu (viz kapitola 5.7 Proces prezentace Výkazu).

5.1.3 Výstup procesu

Výkazy jsou efektivně popsány metadaty tak, aby bylo možné jejich vyplnění Osobami a zaslání do ČNB.

5.2 Proces definice Metodiky vykazovacího rámce

Metodika vykazovacího rámce (viz kapitola 3.1.1 Objekt Metodika vykazovacího rámce) slouží k podpoře práce s Výkazy při tvorbě nového nebo úpravě stávajícího metapopisu jednotlivých Výkazů. Vzhledem k tomu, že na práci s Výkazy se podílí větší množství uživatelů, kteří mají buď zodpovědnost za určité skupiny Výkazů a průřezové změny nebo za správu objektů v Knihovně, je nutné v zájmu podpory organizace práce umožnit jednotlivým uživatelům flexibilně provádět úpravy Výkazů, za které nesou v daném okamžiku zodpovědnost (viz Obrázek 20 - Proces definice Metodiky vykazovacího rámce). V jedné Metodice vykazovacího rámce může pracovat více uživatelů v zájmu zastupitelnosti. Prováděné úpravy objektů resp. tvorba nových je vždy evidována na konkrétního uživatele (viz dokument F – Uživatelé a přístupová práva, kapitola 2.9.12.1 Typ oprávnění „metadata“).



Obrázek 20 - Proces definice Metodiky vykazovacího rámce

5.2.1 Spouštěč procesu

Proces je spouštěn v případě, že analýza potřeb (mimo systém SDAT) ukázala potřebu:

- úpravy stávajících Výkazů,
- vytvoření nových Výkazů,
- ukončení platnosti stávajících platných Výkazů.

5.2.2 Průběh procesu

Proces probíhá v Metodice vykazovacího rámce a je řízen uživatelem. Uživatel (pravděpodobně bude tato funkcionalita na speciální roli) založí Metodiku vykazovací rámce (nutné splnění pravidel – viz kapitola 3.1.1) s datumem, od kdy provedené změny v metodice nabývají platnosti. V nově založené Metodice vykazovacího rámce jsou automaticky poslední verze výkazů z předchozí metodiky (stejného vykazovacího rámce). Na základě věcné analýzy, které výkazy budou upravovány, pak uživatel dotčené výkazy zaverzuje (případně drobných změn je možná pouze varianta). Tyto verze výkazů automaticky přebírají metodickou platnost (platnost_od) od dané Metodiky vykazovacího rámce.

Pokud chce uživatel do nové Metodiky vykazovacího rámce zařadit výkaz, který v dané Metodice vykazovacího rámce není, automaticky se pro daný výkaz vytvoří verze v nové metodice tj. v původní Metodice vykazovacího rámce (např: COREP20150101) má výkaz verzi 2.0 (stav platný), tato verze v původní metodice zůstává. V nové Metodice vykazovacího rámce (např. LIKVID20160101) je v důsledku zařazení automaticky výkaz zaverzován – verze 3.0 (stav projektovaný). Zároveň systém musí zajistit správné zařazení Výkazu do příslušných objektů (Vykazovací rámec, Výkaz ve vykazovacím rámci).

Při další práci systém:

- kontroluje název a kód nově vytvářených Výkazů v proti případným tvorbám duplicit; systém neumožní vytvoření Výkazu s již použitým kódem napříč celým systémem,
- všem uživatelem nově vytvořeným verzím nebo variantám Výkazů nastaví stav Projektovaný,
- po spuštění kontroly konzistence nad verzemi/variantami Výkazů ve stavu Projektovaný, Schválený nebo Platný informuje uživatele o změnách provedených na objektech použitých v těchto verzích/variantách Výkazů. To, zda je pro přijmutí provedených změn nutné vytvořit verzi nebo variantu Výkazu (viz kapitola 2.3 Vazby mezi jednotlivými objekty) je součástí této informace,
- informuje uživatele o změnách provedených na objektech použitých ve verzích/variantách Výkazů zařazených do Metodiky vykazovacího rámce. To, zda je pro přijmutí provedených změn nutné vytvořit verzi nebo variantu Výkazu (viz kapitola 2.3 Vazby mezi jednotlivými objekty) je součástí této informace. Pokud pro přijmutí provedených změn:
    - již existuje v dané Metodice vykazujícího rámce odpovídající verze/varianta Výkazu, Bloku výkazu i Datových oblastí, systém propaguje všechny změny a o provedení této akce informuje uživatele,
    - neexistuje v dané Metodice vykazujícího rámce odpovídající verze Výkazu, Bloku výkazu a Datových oblasti (např. systém pro propagaci změn vyžaduje verzi Výkazu, Bloku výkazu a Datové oblasti, ale uživatel v dané metodice vytvořil

pouze jejich varianty), systém vyzve uživatele k jejich vytvoření. Pokud by je uživatel neudělal, nová varianta Výkazu by při schvalování neprošla kontrolami celkové konzistence (viz kapitola 2.5 Kontrola konzistence),

- systém vede přehled všech verzí a variant Výkazů, které byly zařazeny do jednotlivých Metodik vykazovacího rámce, tento přehled systém uživateli zobrazí ve formě tabulky (gridu),
- systém vede aktuální přehled verzí/variant Výkazů zařazených Metodiky vykazovacího rámce; tento přehled systém uživateli zobrazí ve formě tabulky (gridu), z něhož je možné otvírat jednotlivé Výkazy, se kterými chce uživatel pracovat.

Uživatel může za podpory systému, kdykoliv v průběhu práce v Metodice vykazovacího rámce, provádět následující aktivity:

- vytvoření první verze Výkazů, tj. jejich založením jako objektu (viz kapitola 2.4.5.1 Zakládání instance objektu),
- přesunutí verze/varianty Výkazu do jiné Metodiky vykazovacího rámce jiného Vykazovacího rámce,
- vytvoření nové verze nebo varianty Výkazu a jemu podřízených Bloků výkazu a Datových oblastí,
- smazání verzí/variant Výkazu, resp. jemu podřízených Bloků výkazu a Datových oblastí, které jsou ve stavu Projektovaný,
- spouštění kontroly konzistence (viz kapitola 2.5 Kontrola konzistence),
- schvalování Výkazů hromadně.

5.2.3 Výstup procesu

Je vytvořena Metodika vykazovacího rámce obsahující nové verze/varianty Výkazů, které budou dále upravovány nebo vytvářeny v procesu tvorby výkazů (viz kapitola 5.3 Proces tvorby Výkazu). V případě, že výkaz nebude upravován, je v Metodice vykazovacího rámce předchozí (poslední) verze/varianta. Metodika vykazovacího rámce je dostupná uživateli ze seznamu Metodik vykazujícího rámce. Uživatel má k dispozici přehled Výkazů s vybranými atributy, které mu umožní sledovat postup práce na Výkazech.

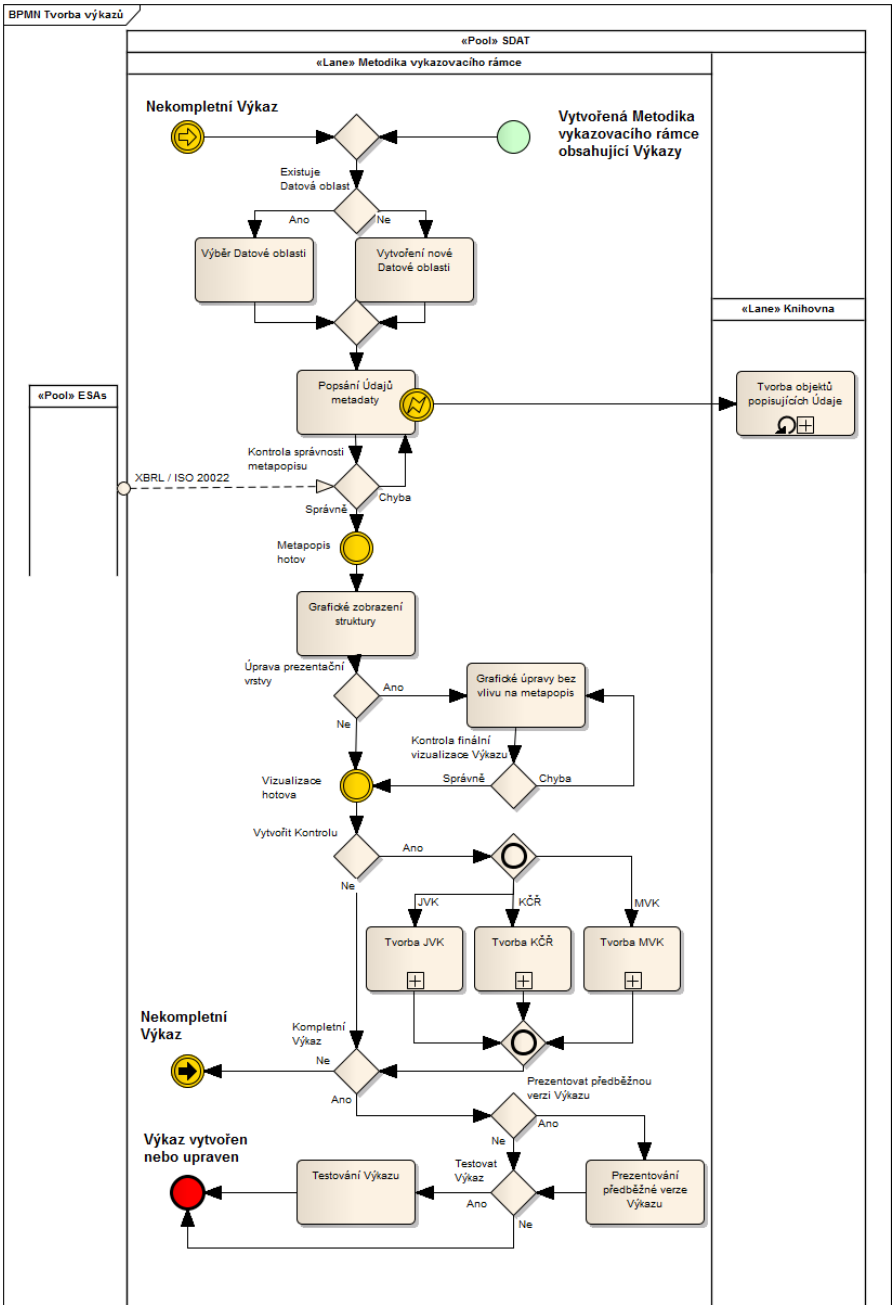
5.3 Proces tvorby Výkazu

Vzhledem k zákonným povinnostem ČNB sbírat data od Vykazujících osob je Tvorba výkazu základní aktivitou tvorby celého metapopisu. Základním stavebním prvkem sběru dat je Údaj (viz kapitola 3.18 Objekt Údaj). Každý Údaj je v systému unikátní, zatímco Ukazatel a Parametry mohou být v systému použity vícekrát, čímž vytváří další jedinečné Údaje. Údaje jsou organizovány v Datových oblastech, Blocích výkazu a Výkazech. Z dosavadních zkušeností ČNB se sběrem dat plyne, že Údaje mohou být organizovány v různých typech Výkazů, a to v závislosti na charakteru a rozsahu sbíraných dat. Datová oblast (viz kapitola 3.4 Objekt Datová oblast) má charakter multidimenzionální tabulky, u níž počet buněk je nebo není předem znám.

Proces tvorby výkazu zahrnuje vytváření nových Výkazů a úpravy stávajících Výkazů. Součástí vytváření metapopisu Výkazů jsou i subprocesy:

- tvorba objektů popisujících Údaje (viz kapitola 5.4 Proces tvorby objektů popisujících údaje),
- tvorba kontrol výkazu (viz kapitola 5.5 Proces tvorba kontrol výkazu).

Atributy používaných objektů jsou popsány v kapitole 2.4.2 Standardní atributy objektů a v kapitolách týkajících se jednotlivých objektů (viz kapitola 3 Objektový model). Životní cyklus používaných objektů je popsán v kapitole 2.4.4 Životní cyklus objektu.



Obrázek 21 - Proces tvorby Výkazu

5.3.1 Spouštěč procesu

Proces je spouštěn na ad hoc bázi v případě potřeby vytvoření nového Výkazu nebo úpravy již existujícího Výkazu. Proces lze realizovat po vytvoření Metodiky vykazovacího rámce.

5.3.2 Průběh procesu

Proces probíhá v Metodice vykazovacího rámce a je řízen uživatelem za podpory systému. Obecně pro proces projektování platí následující:

- upravovat metapopis lze pouze u Výkazů, které jsou ve stavu Projektovaný. Pokud se jedná o úpravu metapopisu Výkazu, jenž byl v minulosti ve stavu Platný, je nutné vytvoření jeho nové verze nebo varianty (viz kapitola 2.3 Vazby mezi jednotlivými objekty),
- v rámci tvorby Výkazu je nejdříve vytvořena instance objektu Výkaz, následně instance objektu Blok výkazu, jenž je do Výkazu zařazena, a nakonec instance objektu Datová oblast, jež je zařazena do Bloku výkazu. Tvorba metapopisu Výkazu se tak odehrává primárně na úrovni Datových oblastí,
- systém umožňuje vytvořit novou Datovou oblast některým z následujících způsobů:
    - prostřednictvím formuláře pro vytvoření Datové oblasti s tím, že uživatel ve formuláři vyplní všechny povinné atributy Datové oblasti,
    - replikací již existující Datové oblasti, tj. uživatel vyhledá Datovou oblast, která je ve stavu Projektovaný, Schválený nebo Platný, vybere verzi/variantu Výkazu, do které požaduje Datovou oblast replikovat, a spustí funkci pro vytvoření repliky. Systém vytvoří identickou Datovou oblast, kterou uživatel může modifikovat,
    - klonováním již existující Datové oblasti, tj. uživatel vyhledá Datovou oblast, která je ve stavu Schválený, Platný případně Projektovaný (viz kapitola 3.4 Objekt Datová oblast), vybere verzi/variantu Výkazu, do které požaduje Datovou oblast klonovat a spustí funkci pro vytvoření klonu. Následně vyplní atribut garant objektu,
- pokud je upravována mateřská Datová oblast, tj. Datová oblast, která má vazbu alespoň na jeden klon (potomek mateřské Datové oblasti), je uživatel na tuto skutečnost systémem upozorněn (uživatel je informován o tom, v jakých Výkazech se dané Datové oblasti vyskytují a kdo je jejich garantem). Před samotným provedením úprav takové Datové oblasti musí uživatel rozhodnout, zda se změny v mateřské Datové oblasti mají projevit v jejich klonech (potomcích mateřské Datové oblasti). Uživatel bude systémem informován o vytvoření nové verze/varianty mateřské Datové oblasti. Těsně před vytvořením verze/varianty mateřské DO (tedy před potvrzením akce) systém uživateli oznámí, že pokud nechce zrušit vazby na klonované DO, musí Výkazy obsahující klonované DO v Metodice vykazovacího rámce, kde je možné projektovat. V opačném případě by systém zrušil vazby klonovaných DO na mateřskou DO s vědomím uživatele. Uživatel má tak následující možnosti:
    - změny v mateřské Datové oblasti požaduje promítnout do vybraných klonů Datové oblasti (potomků mateřské Datové oblasti): v takovém případě musí uživatel mít všechny Výkazy, které obsahují klony Datové oblasti, v kterých požaduje propagovat změnu provedenou v mateřské Datové oblasti, ve stavu Projektovaný. Systém následně provede aktualizaci potomků tak, aby byly totožné s mateřskou Datovou oblastí. To znamená, že se pro Výkaz, kam je klonovaná Datová oblast

zařazena, musí vytvořit nová verze (viz kapitola 2.3 Vazby mezi jednotlivými objekty).
    - změny v mateřské Datové oblasti nepožaduje promítnout do vybraných klonů Datové oblasti (potomků mateřské Datové oblasti): v takovém případě systém zajistí rozpojení vazby mezi mateřskou Datovou oblastí a jejími vybranými klony. Takové klony Datové oblasti se tak stávají zcela nezávislé na své mateřské Datové oblasti (systém zajistí, že u instance objektu Datová oblast, která představuje klonovanou Datovou oblast, je atribut „mateřská Datová oblast“ nastaven na hodnotu „NULL“). Rozpojenou vazbu nelze obnovit,
- pokud je upravována klonovaná Datová oblast, tj. Datová oblast, která má vazbu na mateřskou Datovou oblast, je uživatel na tuto skutečnost systémem upozorněn (uživatel je informován o tom, v jakém Výkazu se mateřská Datová oblast vyskytuje a kdo je jejím garantem). Před samotným provedením úprav takové Datové oblasti musí uživatel rozhodnout, zda požaduje zachovat vazbu na mateřskou Datovou oblast nebo ne. Uživatel má následující možnosti:
    - chce zachovat vazbu klonované Datové oblasti na mateřskou Datovou oblast, tj. nesmí změny provádět v klonované Datové oblasti, ale tyto změny musí provést v mateřské Datové oblasti,
    - nechce zachovat vazbu na mateřskou Datovou oblast, tj. rozhodne se změny provést v klonované Datové oblasti. V takovém případě systém zajistí rozpojení vazby mezi klonovanou Datovou oblastí a její mateřskou Datovou oblastí. Tento klon Datové oblasti se tak stává zcela nezávislý na své mateřské Datové oblasti (systém zajistí, že u instance objektu Datová oblast, která představuje klonovanou Datovou oblast, je atribut „mateřská Datová oblast“ nastaven na hodnotu „NULL“). Rozpojenou vazbu nelze obnovit,
- pracuje se s již vytvořenými objekty umístěnými v Knihovně, tj. pokud potřebné objekty dosud neexistují, vytvoří je (viz kapitola 5.4 Proces tvorby objektů popisujících údaje). Tvorba nových objektů v Knihovně je možná i hromadně (Ukazatele, Číselníky resp. jejich položky, Převodníky resp. jejich položky),
- je umožněno potlačit vykazování některých Údajů s tím, že toto potlačení systém zohlední při další práci s metapopisem a promítne jej do všech navazujících objektů (např. v případě třídimenzionální Datové oblasti do třetí osy do věcných kontrol apod.); potlačení vykazování představuje aktivitu, kterou je zamezeno vzniku Údaje,
- vykazovaným Údajům je uživatelem přiřazen povolený obor hodnot buď na úrovni Údaje, nebo na úrovni Ukazatele (pokud je obor hodnot deklarován na Ukazateli zařazeném v Datové oblasti, platí tento obor hodnot pro všechny Údaje z něj odvozené),
- systém vyhledává duplicity v konkretizaci Údajů:
    - v rámci jednoho Výkazu je každý Údaj jedinečný, tj. v případě, že Výkaz obsahuje více Údajů se stejným metapopisem (viz kapitola 3.18 Objekt Údaj) je postup následující:
        - pokud je tato duplicita vytvořena omylem, je nutné změnit metapopis některého z duplicitních Údajů,
        - pokud je tato duplicita vytvořena záměrně, je nutné sdílet Údaj, tj. zobrazit jeden Údaj na více místech Výkazu; sdílený Údaj se vyplňuje ve Výkazu pouze jednou, přičemž uživatel určí pozici ve Výkazu, kde se tento sdílený Údaj vyplňuje,

  - v rámci více Výkazů může být Údaj popsán stejnými objekty jako již jiný Údaj. V takovém případě má každý z těchto Údajů jedinečný interní identifikátor a je součástí odlišného Výkazu. Tyto Údaje nejsou systémem považovány za duplicitní,
- jsou umožněny hromadné funkce, tj. pokud jsou vkládány shodné atributy, Parametry, Datové typy, potlačování vykazování, obor hodnot Údaje apod. k více Údajům, je možné je vkládat najednou k označenému okruhu objektů (např. v grafickém zobrazení Datové oblasti, výběrem ze seznamu objektů apod.),
- v průběhu celé tvorby výkazu resp. jeho Datových oblastí může uživatel spustit funkci, na základě které se provede kontrola konzistence. Uživatel v rámci spuštění funkce kontroly konzistence má možnost vybrat, zda chce tuto kontrolu provést pouze na vybrané Výkazy, nebo na celý obsah Metodiky vykazovacího rámce,
- věcné kontroly je možné definovat již od okamžiku existence metapopisu a grafického zobrazení všech Datových oblastí, nad kterými se má věcná kontrola provádět; tyto kontroly jsou definovány interaktivně přes grafickou podobu Výkazu (viz kapitola 5.5 Proces tvorba kontrol výkazu),
- kontroly je možné jednotlivě nebo hromadně upravovat (např. v kontrolách funkce „nahradit za“ - nahradit hodnotu Vykazovaného parametru P0019.CZK za hodnotu P0019.EUR, apod.) a hromadně duplikovat/kopírovat do jiných Výkazů (pokud je to z věcného hlediska správné),
- dále je možné jednotlivě nebo hromadně upravovat názvy jednotlivých kontrol, odchylky a stupeň závažnosti,
- v případě změny Datové oblasti nikdy nedochází, po vygenerovaní její nové struktury, ke smazání kontrol. Dopad změn provedených v Datové oblasti na kontroly může být následující:
  - změny se netýkají metapopisu Údajů vstupujících do těchto kontrol: v takovém případě zůstává sémantický tvar kontroly beze změny a systém pro tyto kontroly vygeneruje nový uživatelský tvar,
  - změny se týkají metapopisu Údajů vstupujících do těchto kontrol: v takovém případě je nutné změnit sémantický i uživatelský tvar kontroly. Systém u těchto kontrol v jejich sémantickém tvaru indikuje chyby způsobené změnou metapopisu (např. vyznačí červeně) a uživatel musí provést některou z následujících akcí:
    - edituje sémantický tvar kontroly za účelem odstranění indikovaných chyb,
    - vytvoří celý sémantický tvar kontroly od začátku způsobem uvedeným v kapitole 5.5.2.2 Věcné kontroly vytvářené ručně uživatelem,
- systém na základě nového sémantického tvaru kontrol vytvoří jejich uživatelský tvar,
- uživatel rozhodne, zda chce a v jakém rozsahu prezentovat metapopis jím vybraných Výkazů ve stavu Projektovaný určené skupině uživatelů k připomínkám (viz kapitola 5.7 Proces prezentace Výkazu),
- uživatel rozhodne, zda jím vybrané Výkazy otestuje. Testování probíhá tak, že do Výkazu jsou vyplněna resp. importována testovací data a Výkaz je zaslán jménem testovací Vykazovací osoby nebo vybrané skutečné Vykazovací osoby v testovacím režimu (viz dokument A – Obecné požadavky, kapitola 2.1.2 Testovací prostředí),
- v případě, že jsou dvě verze/varianty jednoho Výkazu ve stavu Projektovaný, systém promítá změny provedené ve verzi/variantě s dřívějším datem platnost_od i do verze/varianty s pozdějším datem platnost_od. V opačném směru tyto změny systém

nepromítá. Pokud je jakýkoliv objekt daného Výkazu změněn ve verzi/variantě Výkazu s pozdějším datem platnost_od, nejsou od okamžiku této změny již žádné následné změny tohoto objektu provedené ve verzi/variantě Výkazu s dřívějším datem platnost_od propagovány do verze/varianty Výkazu s pozdějším datem platnost_od.

5.3.2.1 Formulářové prostředí

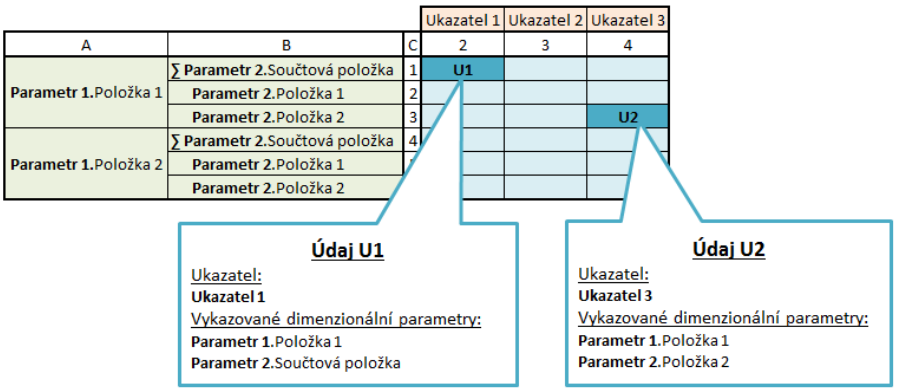
Tvorba výkazu ve formulářovém prostředí je založena na výběru a použití jednotlivých objektů, které popisují požadovaný Údaj. Vychází z formuláře, který umožňuje popsat právě jednu Datovou oblast (viz kapitola 3.4 Objekt Datová oblast). Prostřednictvím tohoto formuláře uživatel přidává Ukazatele (viz kapitola 3.16 Objekt Ukazatel) a Parametry (viz kapitola 3.17 Objekt Parametr) v požadovaném pořadí na jednotlivé osy, případně na části jednotlivých os Datové oblasti, příp. na Datovou oblast jako celek.

Každá osa je nepovinně členitelná na další části (objekt Část osy), která umožňuje deklaraci různých Ukazatelů a oboru hodnot jednotlivých Parametrů pro různé části osy. Pro rozdělení os do částí platí:

- Každý parametr deklarovaný na ose musí být deklarován ve všech jejích částech, pokud osa části obsahuje.
- Ukazatele musí být deklarovány ve všech částech osy, pokud osa části obsahuje.
- Kartézský součin Konkretizovaných Parametrů a množiny Ukazatelů nebude aplikován na celou osu, nýbrž na každou jednotlivou část osy zobrazené jako bloky postupně za sebou.
- V každé části osy může být pořadí (patro) Parametrů a množiny Ukazatelů rozdílné.
- V každé části osy může být obor hodnot Parametrů různý.
- V každé části osy může být množina Ukazatelů různá, vč. jejich hierarchického uspořádání.

Systém umožňuje uživateli ve vstupním formuláři:

- zobrazit objekty použité ve vybrané Datové oblasti; do této Datové oblasti může uživatel přidávat nové objekty nebo z ní odebírat již použité objekty, měnit jejich umístění a pořadí na jednotlivých osách, případně v jejich částech,
- vybírat a používat objekty z Knihovny (viz kapitola 4 Knihovna) bez ohledu na stav jejich verze/varianty formou výběru ze seznamu:
    - Ukazatele s tím, že přímo ve formuláři uživatel určí jejich umístění a pořadí na osách Datové oblasti, případně v jejich částech. V případě, že uživatel pro použité Ukazatele nebo Parametry požaduje jejich hierarchické členění, a to i ve více úrovních, je mu toto umožněno funkcionalitou formuláře na všech osách,
    - Parametry a jejich vazby na Domény číselníků, Hierarchie číselníků a Číselníky, resp. Položky číselníků nebo Položky hierarchií s tím, že přímo ve formuláři:
        - uživatel určí umístění Parametrů a pořadí na osách Datové oblasti, případně v jejich částech. V případě, že vybírané Položky číselníku mají již určenou hierarchii (viz kapitola 3.7 Objekt Hierarchie číselníku), musí být dodržena. Pokud uživatel požaduje vytvoření pořadí více Parametrů použitých na jedné ose, tvořící tzv. patra, je mu toto umožněno funkcionalitou formuláře,
        - umístění Ukazatelů a Parametrů na osách Datové oblasti, případně v jejich částech musí být z grafického zobrazení Datové oblasti patrné (viz Obrázek 22 - Zobrazení umístění Ukazatelů a Parametrů v Datové oblasti),



Obrázek 22 - Zobrazení umístění Ukazatelů a Parametrů v Datové oblasti

  - výběr objektů Knihovny je umožněn v časovém kontextu upravovaného Výkazu (platnost_od) podle jednotlivých atributů objektu, existujících vazeb mezi objekty, případně jejich kombinace,
  - použití Ukazatelů a Parametrů na jednotlivých osách, případně v jejich částech není limitováno a umožňuje vzájemnou kombinaci a variabilní pořadí,
- pro každou osu, nebo část osy zobrazit produkt kartézského součinu Ukazatelů a Konkretizovaných parametrů a pro každý produkt nastavit, zda je daná kombinace Ukazatele a Vykazovaných parametrů chtěná (zobrazí se ve vygenerované struktuře) či nikoliv (taková kombinace nebude ve struktuře zobrazena a nebudou se z ní generovat Údaje). Dále umožní jednotlivých produktům kartézského součinu nastavit jejich pořadí, v jakém budou na dané ose postupně zobrazované,
- spustit funkci transpozice Datové oblasti, tj. „prohodit“ vše, co je deklarováno na jedné ose, a naopak,
- spustit funkci Grafického zobrazení struktury, tj. zobrazení Datové oblasti ve formě tabulky (dvou- nebo třírozměrné); vygenerovaná struktura je věrným zobrazením umístění a hierarchií použitých objektů. Případná hierarchie i ve více úrovních musí být ve struktuře Datové oblasti graficky zobrazena na všech osách,,
- uživatel spustí funkci pro vygenerování Údajů, čímž vzniknou jednotlivé Údaje (viz kapitola 3.18 Objekt Údaj).

V případě, že uživatel u Datové oblasti potřebuje upravit její prezentační vrstvu nad rámec možností struktury Datové oblasti, učiní tak v grafickém zobrazení (viz kapitola 5.3.2.1.1 Grafické zobrazení struktury), která slouží i k nastavení atributů na úrovni jednotlivých Údajů.

Pokud Výkaz obsahuje více Datových oblastí, je pro něj nutné aktivitu opakovat pro každou z nich.

5.3.2.1.1 Grafické zobrazení struktury

Funkcionalita grafického zobrazení struktury je součástí formulářového prostředí a slouží:

- k vizualizaci Datové oblasti z nadeklarovaného metapopisu ve formulářovém prostředí. Grafické zobrazení struktury umožňuje kromě vizualizace DO z deklarovaného metapopisu zobrazit také náhled struktury, která abstrahuje od nastavení chtěných či

nechtěných produktů kartézského součinu na jednotlivých osách a nastavení vlastního pořadí jednotlivých produktů na jednotlivých osách,

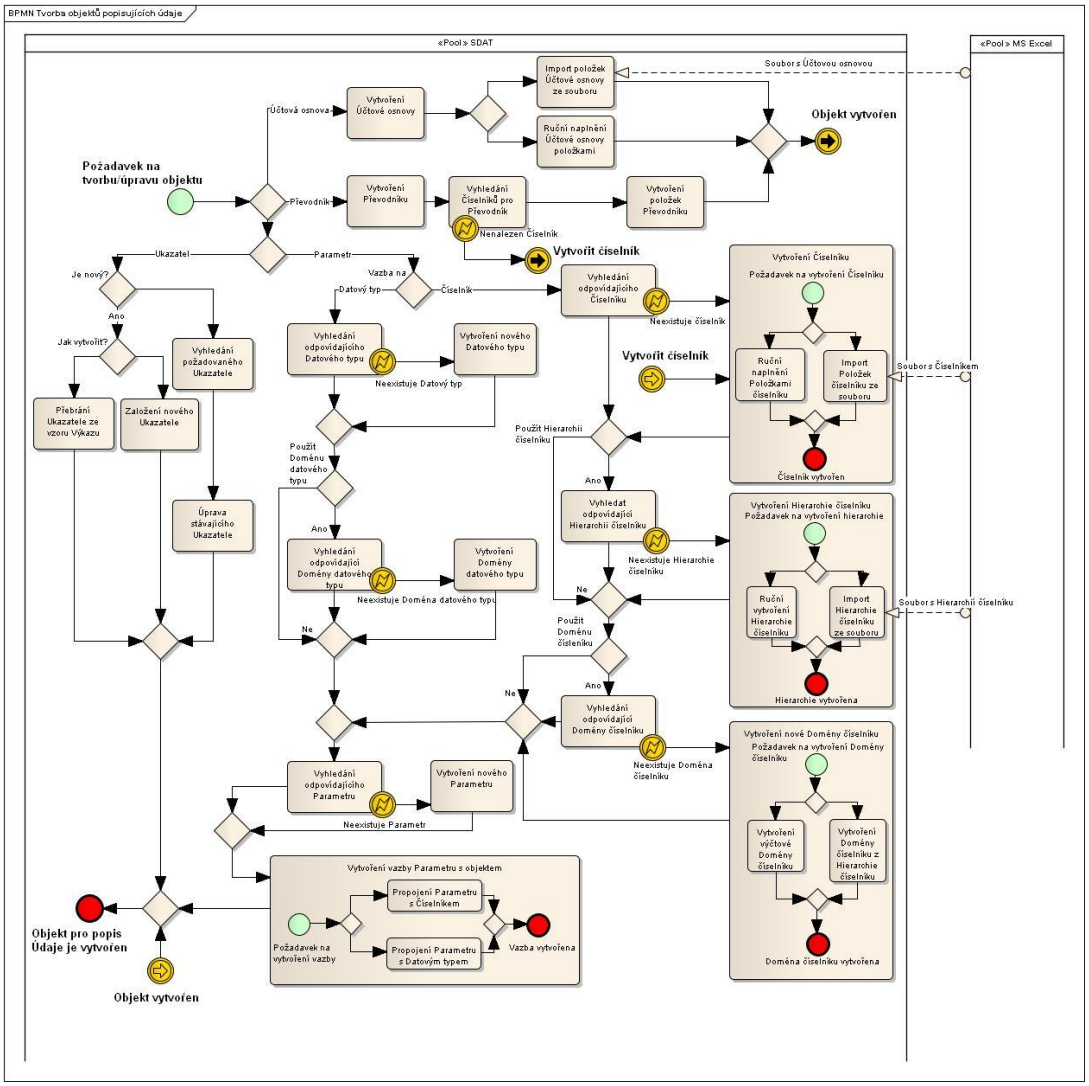
- ke grafickým úpravám, které nemají vliv na metapopis. Uživateli je umožněno přesouvat řádky a sloupce a nastavovat jim jiné pořadí. Tato akce okamžitě změny promítá do příslušného formuláře, kde se tato pořadí nastavují deklarativně. Dále je uživateli umožněno mezi řádky či sloupce vložit textovou informaci, která není tvořena metapopisem, nemá vliv na stávající Údaje Datové oblasti či další generování Údajů.,
- k nastavení hodnot atributům na úrovni jednotlivých Údajů ve struktuře Datové oblasti. Uživateli je umožněno nastavit atributy Údaje, a to:
  - stupeň citlivosti údaje,
  - vykazovat („vykřížkovaný údaj“),
  - zobrazovaný údaj,
- k volitelné úpravě názvů (ukazatelů, parametrů a položek parametrů) objektů, které se v Datové oblasti vizualizují z nadeklarovaného metapopisu. Úprava názvů je platná pouze pro danou Datovou oblast, touto úpravou se nemění názvy objektů v Knihovně. Volitelná úprava názvů objektů v grafickém zobrazení struktury v podstatě upravuje „text ve struktuře“ vizualizované Datové oblasti Upravené názvy jsou pak dále prezentovány Vykazujícím osobám,
- k označení Údaje a hierarchie, pro které se nevygenerují automatické kontroly. Údaje a hierarchie může uživatel označit jednotlivě i hromadně.
- v případě, že uživatel již vygenerovanou strukturu Datové oblasti chce znovu vygenerovat, má možnost si (před daným úkonem) vybrat, zda požaduje:
  - zachování původních názvů objektů, které se ve struktuře zobrazují na základě metapopisu, tedy i zachování volitelné úpravy „test ve struktuře“,
  - zachování údajů, resp. zachování již „vykřížkovaných“ údajů (pokud se nezměnil jejich metapopis.

5.3.3 Výstup procesu

Výstupem je požadovaný Výkaz kompletně popsaný metadaty.

5.4 Proces tvorby objektů popisujících údaje

Základním stavebním kamenem sběru dat je Údaj, jenž je popsán Ukazatelem a jeho konkretizace prostřednictvím Parametrů. K efektivnímu popisu Údaje musí příslušné objekty existovat v systému, a pokud tam žádné vhodné neexistují, musí být vytvořeny. Spolu s Ukazateli a Parametry jsou vytvořeny také jim podřízené objekty a nastaveny vazby mezi nimi.



Obrázek 23 - Proces tvorby objektů popisujících Údaje

5.4.1 Spouštěč procesu

Proces je spouštěn na ad-hoc bázi, tj. v následujících případech:

- uživatel nemá v systému k dispozici všechny potřebné objekty, které by bylo možné použít pro popis Údajů,
- existující objekty popisující Údaje musí být upraveny z legislativních nebo jiných důvodů.

5.4.2 Průběh procesu

Proces je řízen uživatelem a v závislosti na typu objektu (viz kapitola 2.4.5.1 Zakládání instance objektu) je nebo není požadováno, aby návrh uživatele na vytvoření objektu byl potvrzen uživatelem (správcem).

Tvorbu a úpravu objektů popisujících údaje je možné provádět resp. spouštět:

a) z Knihovny, kde je uživateli umožněna tvorba objektů jednotlivě, hromadně, nebo jejich import z externích zdrojů (viz kapitola 6.1 Proces Přebíraní metapopisu z externích zdrojů) resp. jejich změny, např. v případech, kdy změna objektu je vyvolána důvody mimo tvorbu konkrétního Výkazu resp. Datové oblasti (např. vytvoření nebo změna Datového typu, změna Číselníku nebo z něj vytvořené Domény číselníku, úprava atributů jednotlivých objektů apod.),
b) z Metodiky vykazovacího rámce, kde může uživatel vytvářet objekty jednotlivě nebo hromadně).

Tvorba nových objektů popisujících údaje je umožněna (viz kapitola 2.4.5.1 Zakládání instance objektu):

- vytvořením zcela nového objektu,
- replikací existujícího objektu,
- importem objektu z externího zdroje (viz kapitola 6.1 Proces Přebíraní metapopisu z externích zdrojů).

Tvorba nových objektů popisujících údaje, probíhá v následujících variantách:

- objekty vytvářené uživatelem (správcem), tj. Parametr, Účtová osnova, Účet, Převodník, Položka převodníku, Datový typ, Doména datového typu:
    - uživatel (správce) vybere ze seznamu Typ objektu, který požaduje vytvořit,
    - systém zobrazí formulář pro založení požadovaného typu objektu a vyplní kód objektu,
    - uživatel ve formuláři vyplní minimálně atributy název a platnost_od,
        - v případě, že je název pro daný typ objektu již v systému použit, systém informuje uživatele (správce) a ten rozhodne, zda má být tato duplicita v systému vytvořena,
        - ostatní atributy může uživatel vyplnit kdykoliv až do okamžiku schválení objektu,
    - systém uloží vytvořený objekt do Knihovny,
- objekty vytvářené uživatelem bez nutnosti potvrzení uživatelem (správcem), tj. Ukazatel, Doména číselníku:
    - uživatel vybere typ objektu, který požaduje vytvořit,
    - systém zobrazí formulář pro založení požadovaného typu objektu a vyplní kód objektu,
    - uživatel ve formuláři vyplní minimálně atributy název (pokud není název převzat z grafického vzoru) a platnost_od,
        - v případě, že je název pro daný typ objektu již v systému použit, systém informuje uživatele a ten rozhodne, zda má být tato duplicita v systému vytvořena,
        - ostatní atributy může uživatel vyplnit kdykoliv až do okamžiku schválení objektu,
    - systém uloží vytvořený objekt do Knihovny,
- objekty, které se rozdělují na globální a lokální (Číselník, Položka číselníku, Hierarchie číselníku, Položka hierarchie):
    - pro globální objekty je postup následující:
        - uživatel (správce) vybere ze seznamu typ objektu, který požaduje vytvořit,

- systém zobrazí formulář pro založení požadovaného typu objektu a vyplní kód objektu a atribut pro správu nastaví na globální.
- uživatel (správce) ve formuláři vyplní minimálně atributy název a platnost_od
  - v případě, že je název pro daný typ objektu již v systému použit, systém informuje uživatele (správce) a ten rozhodne, zda má být tato duplicita v systému vytvořena,
  - ostatní atributy může uživatel (správce) vyplnit kdykoliv až do okamžiku schválení objektu,
- systém uloží vytvořený objekt do Knihovny,

- pro lokální objekty je postup následující:
  - uživatel vybere ze seznamu typ objektu Číselník, který požaduje vytvořit,
  - systém zobrazí formulář pro založení požadovaného Číselníku a vyplní atributy kód objektu a atribut pro správu nastaví na lokální,
  - uživatel ve formuláři vyplní minimálně název a platnost_od,
    - v případě, že je název pro daný typ objektu již v systému použit, systém nepovolí uživateli instanci objektu s takovým názvem vytvořit,
    - ostatní atributy může uživatel vyplnit kdykoliv až do okamžiku schválení objektu,
  - uživatel vytvoří Položky číselníků, které přiřadí tomuto Číselníku a odešle tento návrh uživateli (správci), přičemž Číselník včetně položek je ve „stavu“ Čeká na potvrzení (atribut objektu Číselník),
  - uživatel (správce) rozhodne o:
    - potvrzení objektu: tj. potvrdí požadavek a systém změní „stav“ objektu ze „stavu“ Čeká na potvrzení na „stav“ Potvrzený (atribut objektu) a informuje uživatele o potvrzení; Uživatel (správce) určí Uživatelské místo, které má oprávnění Číselník modifikovat; objekt v tomto „stavu“ lze použít pro popis Údajů,
    - nepotvrzení objektu: tj. zamítne požadavek a uvede důvod zamítnutí; systém změní „stav“ objektu ze stavu Čeká na potvrzení na „stav“ Nepotvrzený (atribut objektu) a informuje uživatele o nepotvrzení; objekt v tomto „stavu“ nelze použít pro popis Údajů; následně má uživatel možnost:
      - upravit návrh pro vytvoření objektu a opětovně jej odeslat uživateli (správci),
      - smazat objekt,
  - uživatel může do vytvořeného Číselníku přidávat další Položky číselníku,
  - uživatel nad vytvořeným lokálním Číselníkem a jeho Položkami číselníku může vytvářet Hierarchie číselníku a Položky hierarchie.Uživatel je zodpovědný za věcnou i technickou správnost.

Změna již existujících objektů popisujících Údaje[3] je prováděna uživatelem. Změna objektů popisujících Údaje probíhá následovně:

[3]Pro úpravy instance objektů lokální Číselník, jeho Položky číselníku, Hierarchie nad lokálním Číselníkem a její Položky hierarchie již není vyžadované jejich opětovné schválení, tj. workflow se nepoužije.

a) uživatel vybere v Knihovně objekt nebo skupinu objektů, které potřebuje upravit,
b) systém indikuje všechny související objekty, na které má prováděná změna dopad a o těchto objektech informuje uživatele provádějícího změnu, přičemž informace o dopadu změn obsahuje název objektu, kód objektu, garanta objektu a zda musí být dotčenému objektu, pro zohlednění provedených změn, vytvořena nová verze, resp. varianta (viz kapitola 2.3 Vazby mezi jednotlivými objekty),
c) provede požadované změny objektu a zároveň systém zkontroluje, zda navrhovaná změna objektu nevede ke vzniku duplicity. V případě, že je objekt systémem vyhodnocen jako duplicita, má uživatel možnost rozhodnout o jejím ponechání. V takovém případě tento objekt nebude dále systémem považován za duplicitní,
d) promítnutí změn provedených u objektů popisujících Údaje do Datových oblastí je řešeno následovně:
    a. v případě, že uživatel, který změnu provedl, je zároveň garantem dotčených Datových oblastí, tj. Datových oblastí, kde je daný objekt použit:
        - v případě, že jsou verze/varianty Výkazů zařazeny do Metodiky vykazovacího rámce, která je toho času upravovaná, systém informuje uživatele o provedených změnách a zda jejich propagace vyžaduje vytvoření nové verze nebo varianty Výkazu, Bloků výkazu a Datových oblastí. Pokud pro přijmutí provedených změn:
            - již existuje odpovídající verze/varianta Výkazu, Bloku výkazu i Datových oblastí, systém propaguje všechny změny a o provedení této akce informuje uživatele,
            - neexistuje odpovídající verze/varianta Výkazu, Bloku výkazu a Datových oblasti (např. systém pro propagaci změn vyžaduje verzi Výkazu, Bloku výkazu a Datové oblasti, ale uživatel vytvořil pouze jejich varianty), systém vyzve uživatele k jejich vytvoření. Pokud by tak uživatel neudělal, nová varianta Výkazu by při schvalování neprošla kontrolami celkové konzistence (viz kapitola 2.5 Kontrola konzistence),
        - v případě, že nejsou Výkazy zařazené do Metodiky vykazovacího rámce, která je toho času upravovaná, systém informuje uživatele o provedených změnách, a zda pro jejich propagaci je nutné vytvoření nové verze nebo varianty Výkazu a jemu podřízených objektů,
    b. v případě, že uživatel, který změnu provedl, není zároveň garantem dotčených Datových oblastí tj. Datových oblastí, kde je daný objekt použit:
        - systém informuje garanty dotčených Datových oblastí na navrhovanou změnu objektu a vyzve je k vyjádření (viz dokument A – Obecné požadavky, kapitola 2.3 Komunikační modul), přičemž součástí upozornění je mimo samotný návrh změny také seznam dotčených Datových oblastí, u kterých jsou garantem,
        - na základě obdržených vyjádření uživatel provede změnu objektu,
        - v případě, že jsou verze/varianty Výkazů zařazeny do Metodiky vykazovacího rámce, která je toho času upravovaná, systém informuje uživatele o provedených změnách a zda jejich propagace vyžaduje vytvoření nové verze/varianty Výkazu a dotčených Datových oblastí. Pokud pro přijmutí provedených změn:

- již existuje odpovídající verze/varianta Výkazu, Bloku výkazu i Datových oblastí, systém propaguje všechny změny a o provedení této akce informuje uživatele,
- neexistuje odpovídající verze Výkazu, Bloku výkazu a Datových oblasti (např. systém pro propagaci změn vyžaduje verzi Výkazu, Bloku výkazu a Datové oblasti, ale uživatel vytvořil pouze jejich varianty), systém vyzve uživatele k jejich vytvoření. Pokud by tak uživatel neudělal, nová varianta Výkazu by při schvalování neprošla kontrolami celkové konzistence (viz kapitola 2.5 Kontrola konzistence),

o v případě, že nejsou Výkazy zařazeny do Metodiky vykazovacího rámce, která je toho času upravovaná, systém informuje uživatele o provedených změnách a zda pro jejich propagaci je nutné vytvoření nové verze/varianty Výkazu a jemu podřízených objektů,

V rámci tvorby objektů popisujících Údaje systém podporuje následující aktivity:

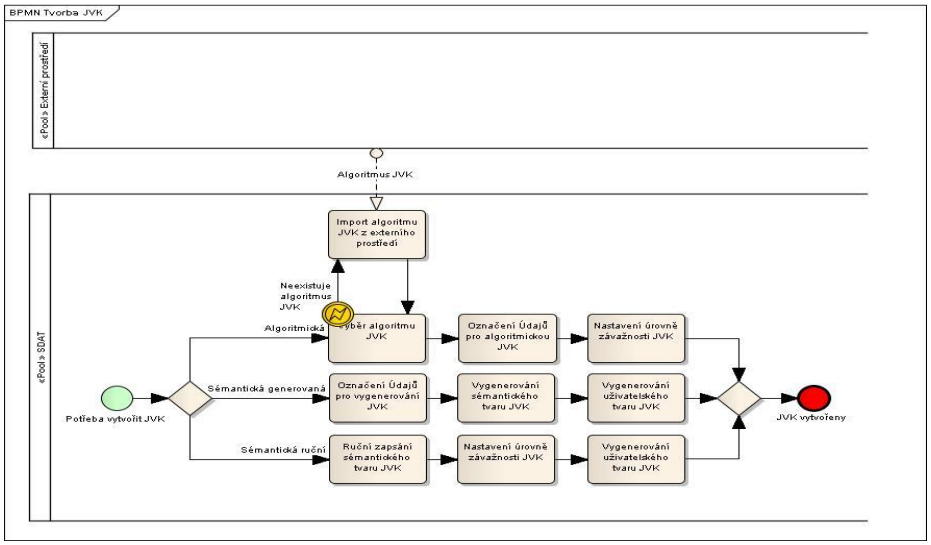
- sleduje případné duplicity v názvech při tvorbě Ukazatelů a Parametrů a upozorňuje na ně uživatele. Na případnou duplicitu je vždy systémem upozorněn uživatel, který daný objekt vytváří, v momentě jeho tvorby,
- vyhledávání instancí objektů stejného typu, které mají použít stejný název objektu. Pokud systém po spuštění této funkcionality objeví instance objektů se stejným názvem, zobrazí je uživateli a ten má možnost tyto instance objektů vzájemně porovnat a rozhodnout o odstranění některé takové instance ze systému,
- komunikace mezi uživateli probíhá v rámci systému (viz dokument A – Obecné požadavky, kapitola 2.3 Komunikační modul),
- uživatel (správce) má možnost zobrazit, revidovat, potvrdit a smazat objekt, který mu připravil jiný uživatel,
- i v případech, kdy je objekt potvrzován uživatelem (správcem), může vlastní úpravy Číselníku provádět jiný uživatel,
- hromadné založení Ukazatelů, Číselníků resp. položek číselníku, Převodníků resp. položek převodníků včetně převzetí některých atributů (např. název),
- systém umožňuje importy atributů popis objektu a poznámka pro Ukazatele, Číselníky a Položky číselníků z externího zdroje,
  - systém podporuje funkci zobrazení a implementace změn do dalších souvisejících objektů, tj. po odsouhlasení Uživatelem provede synchronizaci souvisejících objektů,
  - při výběru z Knihovny se používá výběr ze seznamu (viz dokument A – Obecné požadavky, kapitola 4.6.1 Komponenta Tabulka dat (grid)),
  - systém upozorňuje uživatele na vytváření duplicit v Knihovně,

5.4.3 Výstup procesu

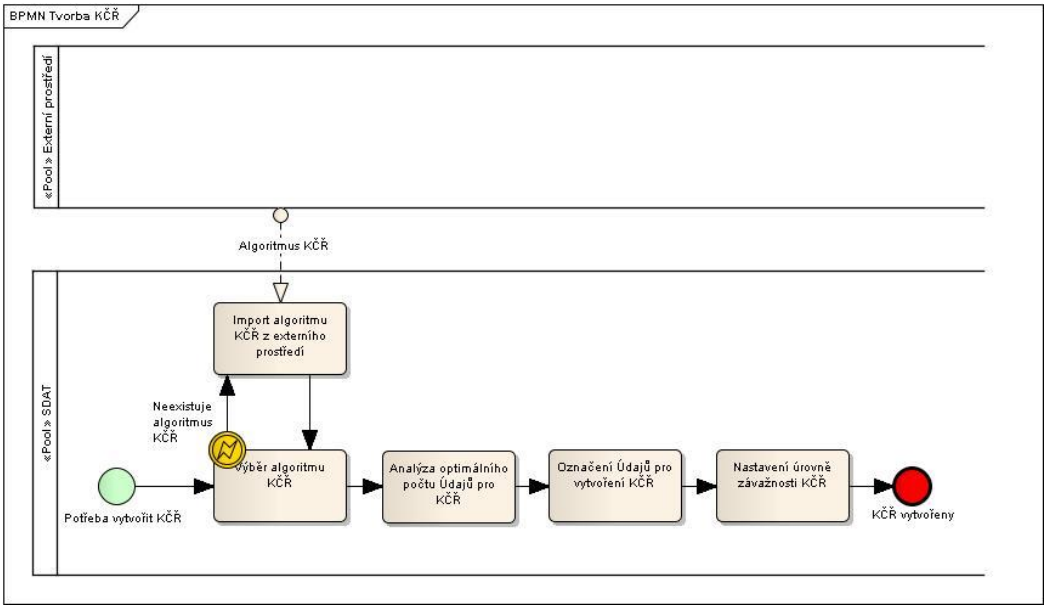
Výstupem procesu je vytvoření všech objektů popisujících Údaje, které jsou potřeba pro tvorbu Výkazů.

5.5 Proces tvorba kontrol výkazu

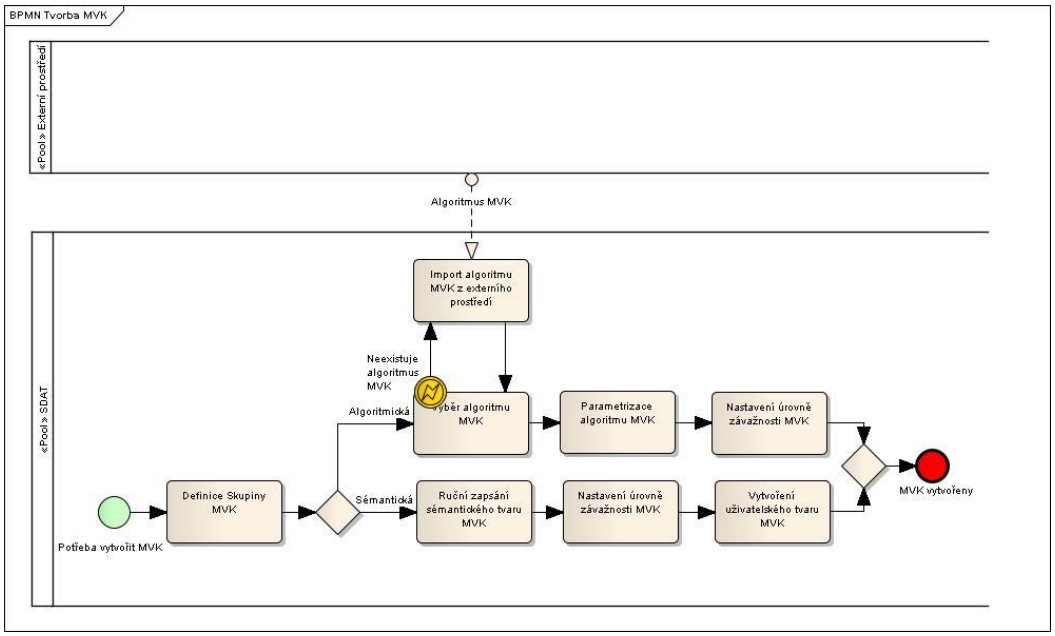
Za účelem zvýšení kvality sbíraných dat je nutné pro vytvořené Výkazy, resp. jejich Datové oblasti definovat kontroly. Tyto kontroly se rozdělují na technické a věcné kontroly. Smyslem těchto kontrol je, aby bylo možné určit, zda přijatá data dosahují požadovanou úroveň kvality a je možné je uložit do databáze ČNB, nebo musí být opraveny. Proces tvorby kontrol výkazu se věnuje pouze věcným kontrolám.



Obrázek 24 - Proces tvorby JVK



Obrázek 25 - Proces tvorby KČŘ



Obrázek 26 - Proces tvorby MVK

5.5.1 Spouštěč procesu

Proces je spouštěn uživatelem v případě, že potřebuje nastavit kontrolní mechanismy mezi Údaji. Pro tvorbu kontrol je nutné, aby byla splněna podmínka, že Údaje, jež se kontroly účastní, musí být popsány metadaty.

5.5.2 Průběh procesu

Proces je řízen uživatelem a v závislosti na stavu Výkazu může probíhat v:

- Metodice vykazovacího rámce, která je toho času upravovaná: kontroly jsou pomocí systému tvořeny nebo upravovány pro Výkazy ve stavu Projektovaný, případně Schválený,
- Metodiky vykazovacího rámce, která není toho času upravovaná: kontroly jsou pomocí systému upravovány pro Výkazy ve stavech Platný; v těchto stavech je možné změnit pouze stupeň závažnosti kontroly, případně stanovit výjimku z definované kontroly (viz dokument C – Vykazovací povinnosti a Registr osob, kapitola 2.5.1 Objekt Výkaz ve Vykazovací povinnosti).

Věcné kontroly je možné definovat:

- automaticky systémem (sémantický tvar),
- ručně uživatelem (sémantický tvar),
- algoritmem (PL SQL).

5.5.2.1 *Věcné kontroly generované automaticky systémem*

Tyto kontroly jsou generovány systémem v sémantickém jazyce na základě hierarchických vztahů definovaných v metapopisu. Z hierarchických vztahů lze odvodit úplný součet (součet Hodnot údajů je roven příslušné součtové položce hierarchie) a neúplný součet (součet Hodnot údajů je menší nebo roven příslušné součtové položce hierarchie). Tímto způsobem se generují pouze JVK v rámci jedné Datové oblasti. Vlastní proces probíhá následovně:

- uživatel vybere Výkaz, resp. jednu nebo více Datových oblastí, pro kterou požaduje vygenerovat kontroly, a spustí funkci pro generování kontrol. V případě, že uživatel nepožaduje vygenerování kontrol ze všech hierarchií definovaných ve Výkazu, má možnost v grafickém vzoru Datové oblasti označit Údaje a hierarchie, pro které se kontroly nevygenerují; tyto Údaje a hierarchie může označit jednotlivě i hromadně,
- systém z hierarchického uspořádání Ukazatelů a konkretizovaných položek Parametrů ve vybrané Datové oblasti vygeneruje kontroly; pokud Datová oblast obsahuje buňky bez metapopisu, systém tyto buňky automaticky nezahrnuje do kontrol; systém vygenerovaným kontrolám nastaví atributy:
    - kód objektu,
    - název objektu, který je odvozený na základě definovaného algoritmu, tento název má možnost uživatel ručně změnit,
    - sémantický tvar včetně odchylky, kterou odvodí na základě algoritmu (viz kapitola 3.19.6 Odchylka v sémantických kontrolách), v závislosti na počtu Údajů vstupujících do kontroly, tuto odchylku má možnost uživatel ručně změnit,
    - úroveň závažnosti kontroly na hodnotu závažná chyba (viz kapitola 3.19.1 Atributy objektu Kontrola), tuto úroveň závažnosti má možnost uživatel ručně změnit,
- systém ze sémantického tvaru kontrol vygeneruje jejich uživatelský tvar.

5.5.2.2 *Věcné kontroly vytvářené ručně uživatelem*

Tyto kontroly jsou definovány uživatelem sémantickým jazykem za pomoci systému (týká se JVK a MVK). Primárně vychází z grafické podoby Datových oblastí, kde uživatel vyznačuje vazby mezi Údaji, nebo hodnotami zaslaných Parametrů u dynamických Údajů (viz kapitola 3.19.3.1 Kontroly vytvořené sémantickým jazykem). Tyto vazby jsou vytvářeny jednotlivě i hromadně. Vlastní proces probíhá následovně:

- uživatel pro Údaje v rámci jedné Datové oblasti nebo mezi více Datovými oblastmi v rámci jednoho nebo více Výkazů ručně vytvoří kontrolu (viz kapitola 3.19 Objekt Kontrola) následujícím způsobem:
    - vybere Výkazy, resp. Datové oblasti, pro které chce definovat kontrolu,
    - systém zobrazí formulář pro vytvoření kontroly a vyplní kód objektu, jehož hodnotu má uživatel možnost změnit,
    - uživatel ve formuláři pro vytvoření kontroly vyplní minimálně atribut název objektu,
    - vybere případně relativní období u MVK, za které dané Výkazy, resp. Datové oblasti do kontroly vstupují, tj. když je požadováno, aby se kontrola prováděla na Hodnotách údajů Výkazů, resp. Datových oblastí vztahujících se k různým

obdobím (např. Hodnota údaje $U1_n$ = Hodnotě údaje $U1_{n-1}$, kde n představuje zvolené období),
    - vytvoří sémantický tvar kontroly:
        - výběrem Údajů z grafického vzoru Datové oblasti, resp. Datových oblastí a přiřazením příslušných funkcí,
        - prostřednictvím průvodce tvorby kontrol, který podle typů Datových oblastí, charakteru a směru opakování kontroly, absolutní a relativní pozice vstupujících Údajů apod., přiřadí vybraným Údajům požadované funkce,
        - přímým zápisem sémantického tvaru kontroly,
        - stanoví požadovanou odchylku,
- vytvořeným kontrolám systém nastaví úroveň závažnosti kontroly na hodnotu „závažná chyba“, přičemž tuto úroveň závažnosti má možnost uživatel ručně změnit,
- uživatel spustí funkci vytvoření uživatelského tvaru kontroly,
- systém správně definovaným kontrolám vytvoří uživatelský tvar a uloží je, přičemž:
    - systém upozorní uživatele na kontroly, které nemají zadanou odchylku,
    - systém pro nesprávně definované kontroly nevytvoří jejich uživatelský tvar a zjištěné chyby v syntaxi indikuje uživateli.

5.5.2.3 Věcné kontroly zapsané algoritmem

Tyto kontroly jsou uživatelem jako již hotové algoritmy v jazyce PL SQL do systému importovány. Algoritmem lze zapsat JVK, KČŘ (včetně MSK) i MVK. Vlastní proces probíhá následovně:

- uživatel vybere Výkaz nebo skupinu Výkazů,
- uživatel ze seznamu algoritmů vybere algoritmus kontroly, který chce na označené Údaje aplikovat; pokud v seznamu algoritmů neexistuje požadovaný algoritmus kontroly, uživatel algoritmus do seznamu naimportuje, tj. algoritmus je vytvořen a kompilován mimo SDAT a importem do SDAT je uložen do seznamu algoritmů pro další využití,
- každý algoritmus, který je uložen do seznamu algoritmů, má vyplněny minimálně tyto atributy:
    - interní identifikátor objektu,
    - název objektu,
    - popis objektu,
- uživatel v rámci vybraného Výkazu nebo skupiny Výkazů vybere Údaj nebo skupinu Údajů, na které má být kontrola aplikována,
- systém vytvoří požadovaný počet algoritmických kontrol a vyplní atributy:
    - kód objektu,
    - název objektu, který je odvozený na základě definovaného algoritmu; atribut název má možnost uživatel ručně změnit,
    - popis objektu, který je přebrán z popisu algoritmu,
    - úroveň závažnosti kontroly na hodnotu „závažná chyba“ (viz kapitola 3.19.1 Atributy objektu Kontrola), přičemž tuto úroveň závažnosti má možnost uživatel ručně změnit,
- systém správně definované kontroly uloží.

Speciálním případem věcných kontrol zapsaných algoritmem jsou kontroly lineární regrese v rámci Kontrol časové řady, u nichž se postupuje následovně:

- uživatel je může definovat pouze pro statické Údaje následujícími způsoby, případně jejich kombinací:
  - spuštěním funkce z grafického vzoru Datové oblasti; systém dle algoritmu označí vhodné Údaje, které jsou účastníky kontrol v časové řadě, toto nastavení uživatel může upravit,
  - označením požadovaných Údajů ve struktuře Datové oblasti,
- uživatel může pro definování kontrol hodnot údajů v časové řadě použít diagnostickou funkci, která na základě počtu vykazovaných Údajů vypočte optimální počet Údajů vstupujících do kontroly lineární regrese. Počet kontrol lineární regrese vytvořených uživatelem se může lišit od počtu navrženého diagnostickou funkcí,
- pro označené Údaje spustí uživatel funkci vytvoření kontrol lineární regrese,
- systém vytvoří požadovaný počet kontrol lineární regrese a vyplní atributy:
  - kód objektu,
  - název objektu, který je odvozen na základě definovaného algoritmu z dotčených dimenzí Údaje,
  - úroveň závažnosti kontroly na hodnotu „chyba k potvrzení“ (viz kapitola 3.19.1 Atributy objektu Kontrola), přičemž tuto úroveň závažnosti má možnost uživatel ručně změnit,
- systém definované kontroly uloží.

5.5.2.4 Obecné požadavky

Obecně v případě kontrol systém dále:

- kontroluje při zadávání kontrol správnost jejich zápisu resp. vyplnění všech potřebných atributů,
- umožňuje automatizované plnění vybraných atributů kontroly,
- podporuje tvorbu názvu kontroly odvozením ze zahrnovaných položek do kontroly podle vybraného algoritmu,
- indikuje kontroly, které musí být prověřeny z důvodu změny metapopisu Údajů vstupujících do kontroly,
- umožňuje hromadné nahrazení použitého Parametru u více kontrol najednou; toto je umožněno pouze v případě, že nahrazovaný Parametr obsahuje stejný výčet Položek číselníku jako nově použitý Parametr,
- umožňuje hromadnou úpravu kontrol (např. funkce „Nahradit za“ – hromadné nahrazení hodnoty Vykazovaného parametru P0019.CZK na P0019.EUR. Funkce platí i pro ostatní objekty, které tvoří zápis kontroly),
- umožňuje hromadné duplikování/kopírování kontrol z výkazu do jiných výkazů,
- umožňuje hromadnou úpravu názvů kontrol, hromadnou úpravu odchylek a úrovně významnosti kontrol,
- umožňuje zobrazovat, přenést do jiných formátů a tisknout všechny kontroly, které se vztahují ke konkrétnímu Údaji, k Datové oblasti, k Bloku výkazu anebo k celému Výkazu, ve volitelném rozsahu (např. pouze uživatelský tvar, sémantický tvar),
- v případě Mezivýkazových kontrol a kontrol mezi více Datovými oblastmi v jednom Výkaze zobrazuje kontroly u všech dotčených Výkazů resp. Datových oblastí resp. Údajů,

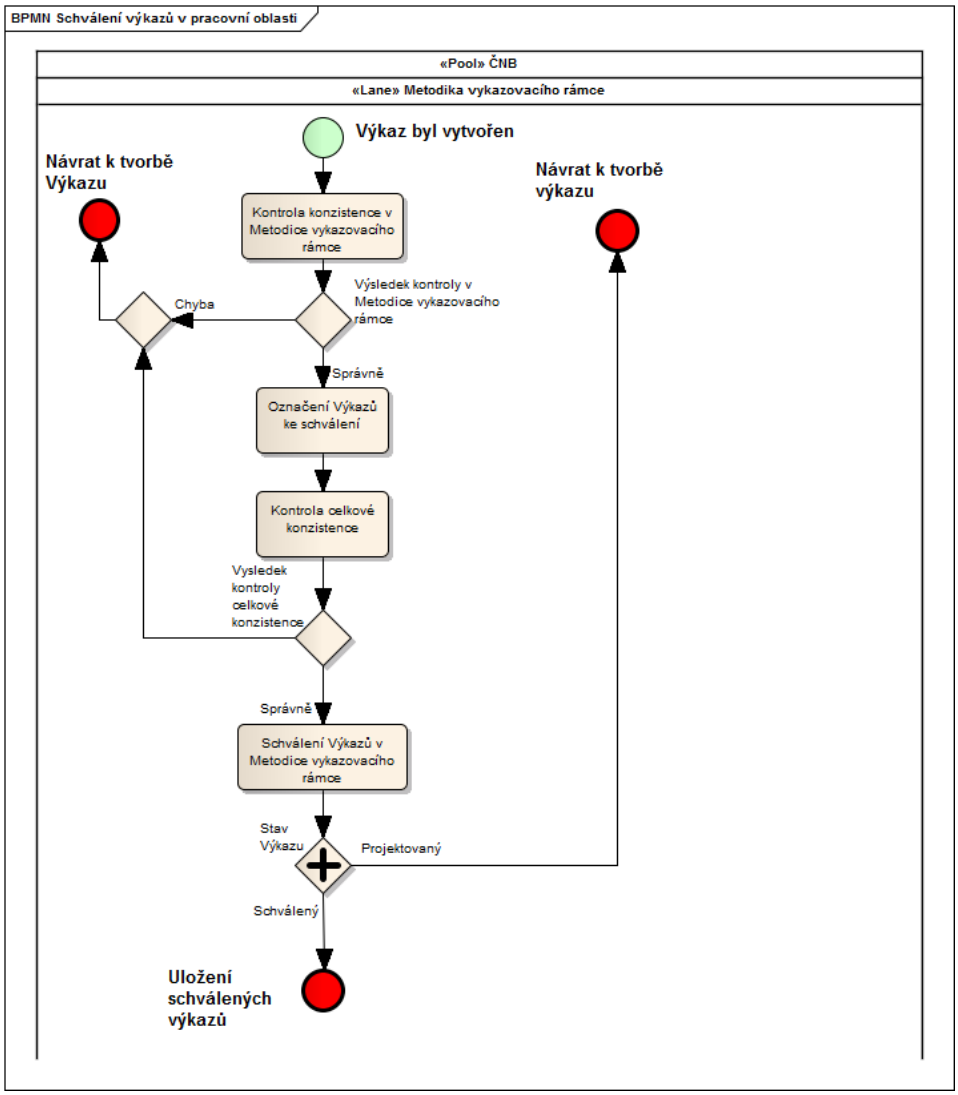
- umožňuje zobrazení konkrétní kontroly vyznačením dotčených Údajů v prezentační vrstvě nebo struktuře Výkazu resp. Datové oblasti, včetně jejich pozice v aplikované funkci,
- provádí řazení kontrol podle Datových oblastí,
- umožňuje uživateli přetřídění kontrol podle jím definovaných kritérií,
- umožňuje přebírat kontroly z externích zdrojů (viz kapitola 6.1 Proces Přebíraní metapopisu z externích zdrojů),
- umožňuje průběžné interní testování nastavení věcných kontrol,
- pro každou kontrolu sestavuje text chybového hlášení pro odesílané protokoly (viz dokument D – Sběr dat, kapitola 2.8.4 Objekt Protokol o zpracování vydání výskytu výkazu),
- umožňuje export kódu PL/SQL algoritmické kontroly do textového souboru,
- umožňuje editaci algoritmů kontrol ze seznamu algoritmů.

5.5.3 Výstup procesu

Výstupem procesu je soubor kontrol výkazu zabezpečující příjem dat v požadované kvalitě. Systém poskytuje souhrnné sestavy s přehledem kontrol k jednotlivým Výkazům a jejich jednotlivým částem.

5.6 Proces schválení výkazů v Metodice vykazovacího rámce

Schválení výkazů v Metodice vykazovacího rámce je finální aktivitou tvorby metapopisu. Součástí tohoto procesu jsou kontroly konzistence (viz kapitola 2.5 Kontrola konzistence) Výkazů v Metodice vykazovacího rámce, jež předchází vlastnímu schválení.



Obrázek 27 - Proces schválení výkazů

5.6.1 Spouštěč procesu

Proces je spouštěn po ukončení tvorby, resp. úpravy Výkazů v dané Metodice vykazovacího rámce. Dané Výkazy nebo jejich část je ve stavu, která umožňuje jejich postoupení k implementaci u Osob.

5.6.2 Průběh procesu

Proces probíhá v Metodiky vykazovacího rámce, která je toho času upravovaná a je řízen uživatelem. Uživatel za podpory systému:

- označí Výkazy určené ke schválení (nebo hromadně označí všechny Výkazy v Metodice vykazovacího rámce a odznačí ty, které ještě nechce schválit, tj. to, co je v daném případě podle počtu Výkazů výhodnější),

- označí Výkazy určené k ukončení platnosti a systém jim nastaví datum platnost_do na hodnotu data platnost_ od Metodiky vykazovacího rámce, která je toho času upravovaná minus jeden den,
- spustí kontrolu konzistence (viz kapitola 2.5 Kontrola konzistence) pro Výkazy vybrané ke schválení,
- systém vybere všechny podřízené objekty, které podléhají schválení a jsou použité ve schvalovaných Výkazech a vyžádá si jejich schválení; pokud schvalovaný Výkaz obsahuje klonovanou Datovou oblast, systém vyžádá rovněž schválení mateřské Datové oblasti (pokud není již ve stavu Schválený nebo Platný). Uživatel následně:
    - schválí ty podřízené objekty, u kterých je autorem (nebo je přiřazen na stejné Uživatelské místo jako autor),
    - u objektů, pro které uživatel není autorem (nebo není přiřazen na stejné Uživatelské místo jako autor), požádá jiného Uživatele, aby tyto objekty schválil (viz dokument A - Obecné požadavky, kapitola 2.3 Komunikační modul). Může nastat následující:
        - uživatel schválí všechny potřebné objekty: v takovém případě lze schválit požadované Výkazy,
        - uživatel neschválí všechny potřebné objekty: v takovém případě nelze schválit požadované Výkazy. Systém informuje uživatele, který si vyžádal schválení objektů, o provedení schválení ze strany jiného uživatele,
- spustí funkci schválení vybraných Výkazů, která se skládá z konečné kontroly konzistence (viz kapitola 2.5 Kontrola konzistence) a vlastního schválení označených Výkazů. Na základě výsledku kontroly konzistence:
    - Výkazy, které neprojdou kontrolou konzistence, nejsou schváleny a jsou zobrazeny uživateli s identifikací zjištěných nedostatků. Stav těchto Výkazu zůstává na Projektovaný,
    - Výkazy, které projdou kontrolou konzistence, jsou schváleny a systém jim nastaví atributy datum schválení na aktuální datum, kdo schválil na uživatele, který spustil funkci schválení, a nastaví stav na Schválený.

5.6.3 Výstup procesu

Výstupem procesu jsou:

- schválené Výkazy a je možné je prezentovat uživatelům.

5.7 Proces prezentace Výkazu

S ohledem na zabezpečení efektivního sběru dat je nutné vytvořené Výkazy prezentovat uživatelům v podobě, která odpovídá všem jejich potřebám. Na straně Osob jsou rozdílné potřeby tzv. business analytiků, kteří jsou odpovědni za mapování vytvořených Výkazů na jejich databáze a rozdílné jsou potřeby IT specialistů odpovídajících za správný přenos dat od Osoby do ČNB. Rovněž uvnitř ČNB se požadavky na prezentaci výkazu liší v závislosti na úkolech jednotlivých uživatelů. Rozdílné požadavky na prezentaci výkazu má uživatel, který se věnuje jeho tvorbě, a jiné požadavky má analytik ČNB, který jednotlivé Údaje mapuje na navazující interní aplikace za účelem dosažení požadovaných výstupů. Sběr dat a prezentace

informací o průběhu sběru dat je součástí dokumentu D – Sběr dat, kapitola 6.14 Monitoring zpracování.

5.7.1 Spouštěč procesu

Proces je spouštěn na ad-hoc bázi v závislosti na typu uživatele a stavu, ve kterém se Výkaz nachází:

- v rámci ČNB je uživateli Výkaz prezentován ve všech stavech v plném rozsahu a plné historii a analytikovi ve stavu Schválený nebo Platný, a to v plném rozsahu a plné historii,
- Osobám je Výkaz prezentován ve všech stavech následovně:
    - ve stavu projektovaném jako předběžný, a to až na základě rozhodnutí uživatele; předběžně prezentovaný Výkaz musí mít dokončen metapopis,
    - ve stavu schváleném a platném v rozsahu a historii definované uživatelem,
        - defaultně je nastaveno, že se Osobám neprezentují následující atributy objektů metapopisu:
            - autor objektu,
            - datum vytvoření objektu,
            - kdo aktualizoval,
            - datum aktualizace,
            - atribut pro správu,
            - vlastnictví,
- veřejnosti je Výkaz prezentován ve stavu Schválený nebo Platný, a to v rozsahu a historii definované uživatelem.

S definovanou hloubkou prezentace Výkazu jsou současně prezentovány objekty Knihovny.

5.7.2 Průběh procesu

Prezentace výkazu probíhá z několika hledisek:

a) **věcného obsahu** prostřednictvím systému znamená:
- zobrazení veškerých atributů objektu vč. odvozených z vlastností objektu a všech objektů s ním souvisejících podle volby uživatele,
- zobrazení metapopisu prostřednictvím všech podporovaných technologií a formátů,
- zobrazení verzí/variant souvisejících objektů v závislosti na vytvořených vazbách,
- grafickou vizualizaci objektů v podobě interaktivních tabulek, hierarchických stromů a seznamů:
    - v interaktivní tabulce zobrazuje grafickou podobu Výkazu a jemu podřízených objektů s tím, že uživateli je umožněno přímo z interaktivní tabulky zobrazovat další objekty závislé na jejich řádcích, sloupcích, kartách a buňkách,
    - v hierarchickém stromu zobrazuje úrovně jednotlivých položek, závislosti mezi nimi a umožněny rozpady jednotlivých větví,
    - v seznamu je možné fulltextově vyhledávat objekty podle jejich názvů, příp. kódů,

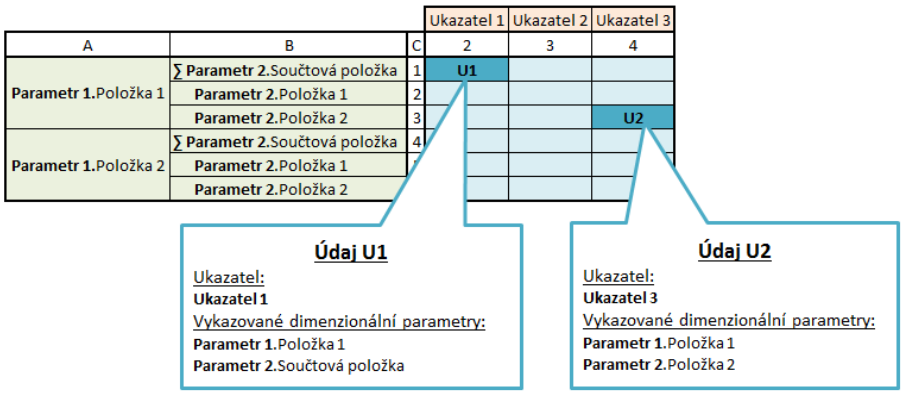
- indikaci a zobrazení změny atributů a verzí/variant závislých objektů v rámci jedné verze/varianty objektu,

b) **rozsahu metapopisu Výkazu;** systém umožní k danému Výkazu nebo skupině Výkazů prezentovat metapopis s ohledem na dopad prováděných úprav:
   - úplný metapopis, tj. kompletní metapopis bez ohledu na to, zda a kdy došlo ke změně metapopisu vybraného výkazu nebo skupiny výkazů nebo vybraných objektů,
   - změnový metapopis, tj. metapopis jen těch částí, kde oproti předcházející verzi/variantě došlo ke změnám. Uživateli bude umožněno, aby si vybral i charakter změn, tj. např. pouze opravy většího charakteru, které vedou ke změně verze, nebo úpravy neovlivňující metapopis, tj. na úrovni variant,

c) podle plánované **technologie a formátu přenosu dat**; systém přeloží výkazy určené k prezentaci do formátů, které jsou použitelné pro přenos dat; do těchto formátů jsou přeloženy i objekty podřízené výkazu; požadovány jsou minimálně formáty:
   - xml,
   - xbrl (v případě formátu XBRL systém nepřekládá, ale nabízí referenci na odpovídající XBRL taxonomii vytvořenou mimo systém SDAT),
   - html.

d) podle **stavu výkazu** resp. objektu (Projektovaný, Schválený, Platný),

e) podle **zvolené grafické podoby** Výkazu, Bloku výkazu nebo Datové oblasti: tj. uživatel má možnost si zvolit zda chce zobrazit pro Výkaz, Blok výkazu nebo Datovou oblast jeho:
   - **strukturu:** struktura Výkazu, Bloku výkazu nebo Datové oblasti je grafické zobrazení objektu, které plně odpovídá vytvořenému metapopisu Údajů (tj. na osách objektu jsou zobrazeny pouze objekty Knihovny, které uživatel určil jako dimenzionální – Obrázek 28 – Ilustrativní zobrazení struktury Datové oblasti). V Grafickém zobrazení struktury lze dále strukturu Datové oblasti upravovat, aniž by to mělo jakýkoliv dopad na metapopis dané Datové oblasti (Obrázek 29 – Ilustrativní zobrazení Datové oblasti s vloženými nemetapopisnými prvky).



Obrázek 28 – Ilustrativní zobrazení struktury Datové oblasti

Obrázek 29 – Ilustrativní zobrazení Datové oblasti s vloženými nemetapopisnými prvky

f) podle **rozhraní:** tj. podle toho, kde je metapopis jednotlivým uživatelům prezentován. Z tohoto pohledu může být metapopis prezentován pomocí:
   - interní aplikace systému SDAT: umožňuje prezentaci výkazu uvnitř ČNB prostřednictvím uživatelského náhledu; prezentovaný metapopis může být exportován do běžně dostupných elektronických formátů (xls, xlsx, doc, docx),
   - komunikačních kanálů: tj. metapopis je Osobám poskytnut ve vybraném formátu sloužícím k přenosu dat (xml, xbrl, html),
   - webové aplikace systému SDAT: umožňuje prezentaci výkazů ve stavu Schválený a Platný v historii definované ze strany ČNB pro:
     - registrované Osoby po přihlášení; vybraným Osobám je možné prezentovat Výkazy i ve stavu Projektovaný za účelem testování a konzultací (viz dokument A – Obecné požadavky, kapitola 2.1.2 Testovací prostředí),
     - veřejnost.

Obecně pro prezentaci Výkazu platí, že:

- umožňuje zobrazit jeden nebo více Výkazů najednou; výběr se provádí:
  - podle vybraného typu nebo subtypu Vykazující osoby,
  - zadáním konkrétní Osoby, tj. Vykazující nebo Zastupující osobě se po přihlášení do systému prezentují všechny Výkazy, které je povinna předkládat,
  - výběrem konkrétního Výkazu přímo nebo ze seznamu,
- respektuje umístění a hierarchickou skladbu použitých objektů,
- umožňuje přímé zobrazení všech objektů, které jsou ve Výkazu použity, resp. ze kterých jsou odvozeny (tj. např. Číselníky, Hierarchie číselníku, Domény číselníků apod.),
- zobrazuje ke každému poli Výkazu veškeré metodické informace k němu se vztahující,
- zobrazení vybraného objektu se provádí z okruhu uživatelského rozhraní, ve kterém se daný objekt nachází,
- je umožněno metapopis všech objektů vytisknout, resp. převést do jiných formátů, které umožňují jejich úpravy, resp. použití pro další účely (xls, xlsx, doc, docx),
- u tisku, resp. exportu jednotlivých objektů lze volit míru podrobnosti (seznam, detailní metapopis Údaje, Číselník s jeho popisem a bez popisu apod.),
- jazyk prezentace je závislý na zvolené lokalizaci a lze jej měnit,
- lze zobrazovat i historické verze/varianty objektů, tj. metapopis vybraných Výkazů a všech souvisejících objektů k danému časovému okamžiku,
- prezentace Výkazu je k dispozici způsobem umožňujícím dálkový přístup,

- umožňuje pro Výkaz zobrazit jeho prezentační vrstvu i jeho strukturu. Obě formy je možné exportovat do formátů (xls, xlsx, doc, docx) a vytisknout.

5.7.3 Výstup procesu

Výkazy jsou uživatelům zobrazeny v podobě a rozsahu, které odpovídají jejich potřebám a rolím.

6 Podpůrné procesy

6.1 Proces Přebíraní metapopisu z externích zdrojů

Objekty metapopisu, které jsou přebírány do SDAT z vnějších zdrojů, vznikají na straně externích autorit (např. EBA, EIOPA, ESMA). Tento metapopis má formu Data Point Modelu (DPM) a z něj vytvořené XBRL taxonomie, nebo je dán XML schématem podle metodiky ISO20022 nebo proprietárním XML schématem navrženým pro specifickou oblast (např. AIFMD). Vzhledem k tomu, že formát XBRL taxonomie, na rozdíl od XML schématu, komplexně pokrývá definici údajů, je v rámci systému SDAT poptávána plná automatizace procesu vytvoření údajů na základě XBRL taxonomie. Přebírání informací z XML schématů naopak není předmětem zadání a bude probíhat nástroji a postupy mimo SDAT. Výstupy z tohoto procesu budou transformovány do podoby standardních souborů pro import metadat a načítány do systémů SDAT manuálně, kde budou využity pro projektování výkazů (např. číselníky a jejich položky, datové typy apod.).

6.1.1 Přebírání metapopisu z XBRL taxonomie

Obsah XBRL taxonomie lze rozdělit do těchto kategorií:

- **Knihovna**– obsahuje metadatové objekty, pomocí kterých jsou popisovány údaje.
- **Definice údajů a vyšších celků, v kterých jsou údaje organizovány** – jednotlivé údaje jsou popisovány pomocí metadat objektů knihovny a organizovány do celků typu výkaz, datová oblast.
- **Struktura výkazu** – obsahuje informaci, jak jsou údaje umístěny do prezentační řádko-sloupcové struktury (x,y,z), popisky a kódy souřadnic os.
- **Věcné kontroly** – obsahují výrazy pro kontrolu obsahu údajů a vztahů mezi nimi.

Předmět procesu přebírání procesu metapopisu z XBRL je automatické načítání všech výše uvedených kategorií metadat do systému SDAT za účelem vytvoření odpovídajících údajů a souvisejících instancí objektů (Výkaz, Datová oblast atd.), které budou následně tímto systémem sbírány prostřednictvím XBRL instančních souborů. Zároveň je možné vykázat tato data i prostřednictvím formátu XML-SDAT.

Přebírání metapopisných objektů musí počítat se změnami, které jsou publikovány formou verzí XBRL taxonomie nebo v případě kontrol pomocí souborů ve formátu excel mimo strukturu taxonomie. Z těchto důvodů musí systém SDAT umožňovat import taxonomie jako:

- plný import celého balíku taxonomie,

- rozdílový import (týkající se pouze změněných nebo nových položek).

Cíle:

Poptávaným řešením je vytvoření jednoho metapopisu založeném na metapopisných objektech SDAT, ale naplněným obsahem z externího zdroje s možností:

- doplnit hodnoty některých atributů, které nejsou obsaženy v externím zdroji, např. česká jazyková verze textů,
- strukturovaně uchovávat informace, které jsou nad rámec vlastností objektů metapopisu[4], např. statický řádko-sloupcový identifikátor nebo reference na instance objektů DPM z XBRL taxonomií.

V oblasti kontrol, kromě jejich načtení z taxonomie a jejich převodu, tak aby byly systémem SDAT interpretovatelné, je možné:

- načíst kontroly publikované mimo XBRL taxonomii (např. EBA excel),
- prostředky SDAT vytvořit kontroly, které jsou nad rámec kontrol vytvořených importem taxonomie,
- snížit úroveň závažnosti kontrol importovaných z taxonomie.

Metapopis importem vytvořených údajů je daný zdrojovou taxonomií a to včetně organizace těchto údajů do celků jako Výkaz, Datová oblast atd. Základní mapování objektů metapopisu ČNB a objektů DPM je následující:

- Modul odpovídá Výkazu,
- Skupina tabulek odpovídá Bloku výkazu,
- Tabulka odpovídá Datové oblasti,
- Domain odpovídá Číselníku,
- Dimension odpovídá Parametru číselníku,
- Member odpovídá Položce číselníku,
- Metric odpovídá Ukazateli,
- Data Point odpovídá Údaji.

Pro metapopis údaje je použito originální kódování z taxonomie, např. je použit načtený číselník a jeho položky. Tímto dochází k opuštění principu praktikovanému v současném systému MtS v podobě jednotných číselníků napříč různými oblastmi výkaznictví. V systému tak bude např. existovat číselník vytvořený a spravovaný metodiky ČNB (BA0010 Kódy měn ISO) a zároveň importem taxonomie EBA vznikne další číselník (EBA_CU). Z důvodu udržení vztahu mezi číselníky pocházejících z různých zdrojů bude prováděno mapování odpovídajících číselníků a jejich položek mezi sebou. Tyto informace budou dostupné i vykazujícím osobám.

Součástí importu je rovněž struktura výkazu (jednotlivých tabulek), která je do SDAT ukládána v originální podobě. Tím, že struktura vzniká v evropských dohledových agenturách odlišným procesem a na základě volnějších pravidel než v SDAT, není požadována funkcionalita, která by umožňovala pomocí prostředků SDAT její změnu a přegenerování.

[4] V zadání nejsou v dokumentu B-Metapopis uváděny atributy nebo případné další objekty vyplývající z vazby metapopisu ČNB na XBRL taxonomii.

Datové oblasti a Údaje v nich obsažené budou importem taxonomie založeny v souladu s pravidly projektování v systému SDAT. Importované Údaje, které tvoří Datovou oblast a současně nesou různý počet Parametrů, budou obohaceny do plné množiny Parametrů Údajů v dané DO. Jako hodnota těchto doplněných Parametrů bude vždy použita implicitní Položka příslušného Číselníku.

S ohledem na stabilitu řešení níže uvedeného procesu je nutné uvést fakt, že DPM není v době vytváření tohoto dokumentu ukotveno žádnou specifikací a existuje pouze jeho model na konceptuální úrovni (popis zde: http://www.wikixbrl.info/index.php?title=European_Data_Point_Methodology). Oficiální technický standard pro přenos metapopisu představuje XBRL taxonomie. Vztah DPM a XBRL taxonomie včetně architektury XBRL taxonomií používaných evropskými dohledovými autoritami je popsán zde: http://www.xbrlwiki.info/index.php?title=European_XBRL_Taxonomy_Architecture_V3.0.

DPM ve formátu MS Access (ACCDB), který je v ČNB používán pro přebírání metapopisu z EBA, je tak pouze jednou z implementací konceptuálního modelu DPM. Relační databázové schéma uchovávající metapopis se může měnit. Stejně tak se od sebe mohou lišit technické implementace DPM mezi jednotlivými autoritami. Při realizaci funkcionality pro přebírání dat je třeba brát na tento fakt zřetel.

Dosud známé vzory XBRL/DPM metapopisu vytvářeného externími autoritami jsou k dispozici na následujících odkazech:

- https://eiopa.europa.eu/regulation-supervision/insurance/reporting-format
- https://www.eba.europa.eu/regulation-and-policy/supervisory-reporting/

6.1.2 Spouštěč procesu

Proces je spouštěn uživatelem na základě uvolnění nové verze XBRL taxonomie na straně externí autority.

6.1.3 Průběh procesu

Proces je řízen uživatelem a jeho průběh je následovný:

- uživatel stáhne verzi/variantu XBRL taxonomie, případně související soubory uvolněné externí autoritou a uloží je na určené místo v ČNB,
- následně prostředky systému SDAT provede jejich import do systému SDAT, kde se naplní příslušné objekty v odpovídajících verzích instancí.

6.1.4 Výstup procesu

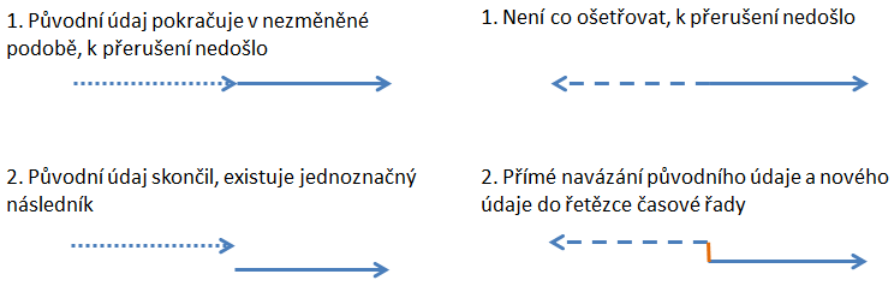
Výstupem procesu je kompletní sada výkazů pokrývající obsah XBRL taxonomie.

6.2 Proces Navazování časových řad Údajů

Časové řady se navazují například v případě, pokud je v nové verzi/variantě Výkazu změněn metapopis Údaje a jeho význam se nemění. Změnou popisu Údaje je v databázi vytvořen nový Údaj (viz kapitola 3.18 Objekt Údaj). Účelem navazování časové řady Údajů je vytvoření vazby mezi více Údaji, které popisují stejnou ekonomickou veličinu tak, aby bylo možné vybírat data v historii delší než je platnost jednoho Údaje.

Navazování časových řad Údajů spojuje původní Údaj („předchůdce“) a nově vzniklý Údaj („následník“) do řetězce, který reprezentuje stejnou ekonomickou veličinu. Takto navázané časové řady jsou dostupné prostřednictvím Modulu pro výběr dat a Uživatelských pohledů (viz dokument E – Výběry dat, kapitola 4 Modul pro výběr dat a kapitola 5 Uživatelské pohledy). Prostřednictvím určité verze/varianty Výkazu lze tedy vybírat data z databáze i za jiné stavy ke dni, než ke kterým se daná verze/varianta Výkazu vztahuje.

Pokud se v nové verzi/variantě Výkazu nezmění metapopis Údaje, časová řada Údaje je zabezpečena a není ji potřeba navazovat, a to ani v případě, kdy se změní pozice Údaje ve struktuře výkazu anebo Údaj přejde do jiného Výkazu.



Obrázek 30 - Navazování časových řad Údajů

6.2.1 Spouštěč procesu

Proces je spouštěn uživatelem v případě, že došlo ke vzniku nového Údaje, který vyjadřuje stejnou ekonomickou veličinu jako již existující Údaj v minulosti a zároveň uživatel požaduje vývoj této ekonomické veličiny v historii.

6.2.2 Průběh procesu

Proces je řízen uživatelem a jeho průběh je následovný:

- uživatel vybere Výkaz, který obsahuje nově vzniklý Údaj nebo více Údajů, pro které je požadováno navazování časové řady,
- systém zobrazí grafický vzor vybraného Výkazu,
- uživatel v grafickém vzoru označí Údaje – následníky,
- uživatel vybere Výkaz, který obsahuje Údaje - předchůdce a jehož platnost_do je menší než platnost_od výše vybraného Výkazu,
- systém zobrazí grafický vzor vybraného Výkazu,

- uživatel v grafickém vzoru označí Údaje – předchůdce a spustí funkci navázání časových řad Údajů,
- systém provede kontrolu splnění následujících podmínek:
  - časová řada je jednoznačná,
  - Údaj - následník není obsažen ve Výkazu (jeho verzi/variantě), jehož interval platnost_od a platnost_do se překrývá s intervalem Výkazu, ve kterém je obsažen Údaj – předchůdce,
- pokud kontroly nejsou v pořádku, navázání časové řady se neuskuteční a systém upozorní uživatele na důvod nenavázání časové řady,
- pokud kontroly jsou v pořádku, uživatel potvrdí připravenou časovou řadu.

Pro navazování časových řad platí obecně:

- navazování časových řad je nepovinné, záleží na příslušném uživateli, zda navázání časové řady provede či nikoliv,
- systém nevyžaduje navázání časové řady při nezměněném metapopisu, tj. pokud ve Výkazu, resp. Datové oblasti nevznikly nové Údaje, časová řada je tak automaticky zajištěna systémem,
- navazování časových řad uživatel provádí prostřednictvím průvodce,
- časové řady lze navazovat pro Výkazy ve všech stavech (Projektovaný, Schválený i Platný),
- pokud mají údaje navázanou časovou řadu do minulosti, systém podporuje výstupy v odpovídajících hloubkách:
  - při výběru hodnot údajů prostřednictvím modulu pro výběr dat a uživatelských pohledů lze prostřednictvím vybrané verze/varianty Výkazu zobrazit Hodnoty údajů ze všech stavů ke dni, které odpovídají délce časové řady Údaje,
  - modul pro výběr dat poskytuje nabídku výběrového kritéria stav ke dni v rozsahu odpovídajícího časovému úseku nepřerušených nebo navázaných časových řad,
- systém provádí kontroly správnosti navázání časových řad:
  - při vlastním navazování časových řad,
  - na vyžádání spuštěním příslušné funkce,
  - při překlápění výkazů do stavu Schválený, protože jsou součástí kontrol celkové konzistence.

6.2.3 Výstup procesu

Je navázána časové řada mezi různými Údaji, které vyjadřují stejnou ekonomickou veličinu. Tuto časovou řadu je možné zobrazit včetně funkčního výrazu v případě, že je článkem řetězce.

7 Funkční požadavky

7.1 Obecné požadavky pro objekty metapopisu

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
OBE_1.0	Smazání instance objektu, která podléhá sledování stavů	Systém umožňuje uživateli smazat jakoukoliv instanci objektu, která je ve stavu Projektovaný (viz kapitola 2.4.5.4 Smazání objektů). Smazáním instance objektu jsou automaticky systémem smazány i jemu podřízené instance objektů, které jsou ve stavu Projektovaný. Smazání instance objektu je možné pouze v případě, že daná instance objektu není použita v jiných instancích objektů. Smazání instance objektu je možné pouze v Metodice vykazovacího rámce), nebo v okruhu Knihovna.	Závazný	1
OBE_2.0	Ukončení platnosti instance objektu uživatelem, která podléhá sledování historie stavů	Systém umožňuje uživateli ukončit platnost jakékoliv poslední verze/variantě instance objektu, která je ve stavu Platný, změnou atributu platnost_do na požadované datum (viz kapitola 2.4.5.3 Ukončování platnosti objektů). Tato akce je povolena pouze v případě, že neexistují další verze/varianty dané instance objektu, které jsou ve stavu Schválený. Ukončení platnosti poslední platné verzi/variantě instance objektu je možné pouze v případě, že daná instance objektu není použita v jiných instancích objektů. Ukončením platnosti poslední verze/varianty instance objektu jsou automaticky systémem ukončeny platnosti všech podřízených instancí objektů.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Zároveň systém smaže všechny následující verze/varianty ukončované instance objektu, které jsou ve stavu Projektovaný, včetně všech jejich podřízených instancí objektů.		
OBE_2.1	Ukončení platnosti instance objektu systémem, která podléhá sledování historie stavů, v závislosti na zplanění jiné verze/varianty téhož objektu	Systém automaticky ukončí platnost verzi/variantě instance objektu, jehož následující verzi/variantě byl uživatelem změněn stav na ze stavu Schválený na stav Platný. Systém nastaví atribut platnost_do verzi/variantě ukončované instance objektu jako datum platnost_od následné verze/varianty instance objektu mínus jeden den. Ukončením platnosti instance objektu jsou automaticky systémem ukončeny platnosti i jemu podřízených instancí objektů, kterým byla zplatněna následná verze/varianta. Systém může takto ukončit platnost pouze instancím objektu, které jsou ve stavu Platný a zároveň nejsou použity v jiných instancích objektů. Podrobně je tento způsob zplatnění jedné verze/varianty a dopad na existující platnou verzi/variantu popsán v kapitole 2.2.6 Přístup „Sledování historie – časová platnost + stavy“.	Závazný	1
OBE_3.0	Prodloužení platnosti instance objektu, která podléhá sledování historie stavů	Systém umožňuje uživateli prodloužit platnost instance objektu, která je ve stavu Platný, změnou atributu platnost_do na datum vyšší než je původní datum platnost_do (viz kapitola 2.4.5.5 Prodloužení platnosti objektu). Prodloužením platnosti instance objektu jsou automaticky systémem prodlouženy platnosti i všem jemu podřízených instancí objektů. Prodloužení platnosti instance objektu je umožněno pouze v případě, že aktuální datum je nižší než datum platnost_do dané instance objektu.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Prodloužení platnosti instance objektu je možné v okruhu Knihovna a okruhu Metodika vykazovacího rámce.		
OBE_4.0	Schválení instance objektu, která podléhá sledování historie stavů	Systém umožňuje uživateli schválit instanci objektu, která je ve stavu Projektovaný, pouze za předpokladu, že kontrola celkové konzistence (viz kapitola 2.5 Kontrola konzistence) provedená nad touto instancí objektu skončila bez závažných chyb a případně všechny v ní použité instance objektů knihovny jsou ve stavu Schválený nebo Platný. Schválením instance objektu systém automaticky schválí všechny jemu podřízené instance objektů, které jsou ve stavu Projektovaný. Schválením instance objektu systém změní stav schvalované instance objektu ze stavu Projektovaný na stav Schválený.	Závazný	1
OBE_5.0	Zplatnění instance objektu, která podléhá sledování historie stavů	Systém umožňuje uživateli zplatnit instanci objektu, která je ve stavu Schválený, pouze za předpokladu, že kontrola celkové konzistence (viz kapitola 2.5 Kontrola konzistence) provedená nad touto instancí objektu skončila bez závažných chyb a všechny v ní použité instance objektů knihovny jsou ve stavu Platný. Zplatněním instance objektu systém automaticky zplatní všechny jemu podřízené instance objektů, které jsou ve stavu Schválený (situace, kdy by nějaký podřízený objekt byl ve stavu Projektovaný, nemůže nastat). Zplatněním instance objektu systém změní stav zplatňované instance objektu ze stavu Schválený na stav Platný. Zároveň musí dojít k ukončení stavu Platný předcházející verze/varianty (pokud existuje), viz OBE_2.1.	Závazný	1
OBE_6.0	Změna stavu instance objektu ze stavu	Systém umožňuje uživateli změnit stav instance objektu ze stavu Schválený na stav Projektovaný, pokud daná instance objektu není již použita v jiné instanci objektu, která je ve stavu Schválený.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	Schválený na stav Projektovaný (u instancí, které podléhají sledování historie stavů)	Systém zároveň změní stav všech jeho podřízených instancí objektů, které jsou ve stavu Schválený na stav Projektovaný. Změna stavu instance objektu ze stavu Schválený na stav Projektovaný je možná v okruhu Knihovna nebo v Metodice vykazovacího rámce, kde je možné aktuálně projektovat.		
OBE_7.0	Změna varianty na verzi	Systém umožňuje uživateli změnit vytvořenou variantu instance na verzi instance objektu (viz kapitola 2.2.1 Číslo verze a varianty instance objektu). Změna varianty instance objektu na verzi této instance objektu je možná, pouze pokud varianta instance objektu je ve stavu Projektovaný.	Závazný	1
OBE_8.0	Nastavení atributu garant	Systém defaultně do atributu garant instance objektu vyplní identifikaci uživatele, který instanci objektu vytvořil. Systém umožňuje uživateli změnit defaultní nastavení výběrem jiného garanta ze seznamu zaměstnanců ČNB.	Závazný	1
OBE_9.0	Vytvoření nové verze/varianty objektu Projektovaný (u instancí, které podléhají sledování historie stavů)	Systém umožňuje vytvořit instanci (verzi/variantu) objektu za dodržení pravidel v kapitole 2.2.1 Číslo verze a varianty instance objektu.	Závazný	1
OBE_10.0	Smazání instance objektu, která podléhá sledování	Systém umožňuje uživateli smazat jakoukoliv instanci objektu, která nemá vazbu na žádnou instanci objektu, jenž je ve stavu Schválený nebo Platný (viz kapitola 2.4.5.4 Smazání objektů).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	časové platnosti	Smazání instance objektu je možné pouze v Metodice vykazovacího rámce, kde je možné aktuálně projektovat, nebo v Knihovně.		
OBE_11.0	Ukončení platnosti instance objektu uživatelem, která podléhá sledování časové platnosti	Systém umožňuje uživateli ukončit platnost jakékoliv poslední verze/variantě instance objektu, která má vazbu na instanci objektu, jenž je ve stavu Platný, změnou atributu platnost_do na požadované datum (viz kapitola 2.4.5.3 Ukončování platnosti objektů). Tato akce je povolena pouze v případě, že neexistují další verze/varianty dané instance objektu, které mají vazbu na instance objektů, jež jsou ve stavu Schválený. Zároveň systém smaže všechny následující verze/varianty ukončované instance objektu, které mají vazbu na instance objektů, jež jsou ve stavu Projektovaný.	Závazný	1
OBE_11.1	Ukončení platnosti instance objektu systémem, která podléhá sledování časové platnosti, v závislosti na zplanění jiné verze/varianty téhož objektu	Systém automaticky ukončí platnost verzi/variantě instance objektu, jehož následující verze/varianta má vazbu na instanci objektu, které byl uživatelem změněn stav na ze stavu Schválený na stav Platný. Systém nastaví atribut platnost_do verzi/variantě ukončované instance objektu jako datum platnost_od následné verze/varianty instance objektu mínus jeden den. Systém může takto ukončit platnost pouze instancím objektu, které mají vazbu na instance objektů, jež jsou ve stavu Platný.	Závazný	1
OBE_12.0	Změna varianty na verzi instance objektu, která podléhá sledování časové platnosti	Systém umožňuje uživateli změnit vytvořenou variantu instance na verzi instance objektu (viz kapitola 2.2.1 Číslo verze a varianty instance objektu). Změna varianty instance objektu na verzi této instance objektu je možná, pouze pokud varianta instance objektu má vazbu na instanci objektu, jenž	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		je ve stavu Projektovaný.		

7.2 Vykazovací rámec

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
VRA_1.0	Vykazovací rámec – nový	Systém umožňuje uživateli založit nový Vykazovací rámec dle jedinečného kódu v systému (viz kapitola 3.1 Vykazovací rámec).	Závazný	1
VRA_2.0	Vykazovací rámec – naplnění Výkazy	Systém umožňuje uživateli naplnit nový Vykazovací rámec vybranými Výkazy. Výkaz musí být v jednom časovém okamžiku vždy pouze v jednom Vykazovacím rámci.	Závazný	1
VRA_3.0	Vykazovací rámec – přesun Výkazů	Systém umožňuje uživateli přesunovat Výkaz mezi jednotlivými Vykazovacími rámci.	Závazný	1
VRA_4.0	Vykazovací rámec - zrušení	Systém umožňuje uživateli zrušit Vykazovací rámec. V okamžiku rušení nesmí Vykazovací rámec obsahovat žádný Výkaz.	Závazný	1
VRA_5.0	Vykazovací rámec – atributy	Systém umožňuje uživateli dle VRA_1.0 nastavení atributů objektu. V závislosti na konkrétním atributu je možné nastavení uživatelem nebo systémem.	Závazný	1

7.3 Metodika vykazovacího rámce

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
MVRA_1.0	Metodika	Systém umožňuje uživateli založit první Metodiku vykazovacího rámce	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	vykazovacího rámce – první metodika	(viz kapitola 5.1.2 Průběh procesu) v případě, že existuje nadřazený objekt Vykazovací rámec. Uživatel nastaví jedinečný kód (viz kapitola 3.1.1 Metodika vykazovacího rámce) a platnost_od dané Metodiky vykazovacího rámce. Platnost_do je nastavena systémem na maximální datum.		
MVRA_1.1	Metodika vykazovacího rámce – nová metodika	Systém umožňuje uživateli založit novou Metodiku vykazovacího rámce (viz kapitola 5.1.2 Průběh procesu) konkrétního Vykazovacího rámce. Uživatel nastaví jedinečný kód (viz kapitola 3.1.1 Metodika vykazovacího rámce) a platnost_od dané Metodiky vykazovacího rámce. Platnost_do je nastavena systémem na maximální datum.	Závazný	1
MVRA_1.2	Metodika vykazovacího rámce – spojité platnosti	Systém hlídá, aby Metodiky vykazovacího rámce jednoho Vykazovacího rámce na sebe spojitě navazovaly. Systém automaticky zkracuje platnost _do předchozí Metodiky vykazovacího rámce, pokud byla založena další Metodika vykazovacího rámce daného Vykazovacího rámce.	Závazný	1
MVRA_1.3	Metodika vykazovacího rámce - obsah	Systém umožňuje uživateli zobrazit obsah Metodiky vykazovacího rámce, obsah verzí použitých objektů. Systém umožňuje hledání a filtrování.	Závazný	1
MVRA_2.0	Metodika vykazovacího rámce – atributy	Systém umožňuje uživateli dle MVRA_1.0 a MVRA_1.1 nastavení atributů objektu. V závislosti na konkrétním atributu je možné nastavení uživatelem nebo systémem.	Závazný	1
MVRA_3.0	Metodika vykazovacího rámce – místo pro	Systém umožňuje uživateli projektovat v konkrétní Metodice vykazovacího rámce dle pravidel projektování a dle procesů (např. kapitola 5.1.2 Průběh procesu).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	projektování			
MVRA_4.0	Metodika vykazovacího rámce – vliv na platnosti verzí objektů	Systém hlídá a automaticky nastavuje platnost_ od verzí objektů, které vznikly v konkrétní Metodice vykazovacího rámce. Platnost_od těchto verzí se řídí atributem platnost_od dané Metodiky vykazovacího rámce (viz kapitola 3.1.1 Metodika vykazovacího rámce).	Závazný	1
MVRA_5.0	Metodika vykazovacího rámce – vznik verzí objektů	Systém umožňuje v Metodice vykazovacího rámce zakládat nové objekty, resp. vytvoření prvních verzí objektů (např. Výkaz, Blok, Datová oblast, atd.), které nespadají do Knihovny. Platnost jednotlivých verzí objektů na sebe musí v systému spojitě navazovat.	Závazný	1
MVRA_5.1	Metodika vykazovacího rámce – modifikace objektů	Systém umožňuje v Metodice vykazovacího rámce modifikovat objekty, (např. Výkaz, Blok, Datová oblast, atd.), které nespadají do Knihovny.	Závazný	1
MVRA_5.2	Metodika vykazovacího rámce – ukončování platností objektů	Systém umožňuje v Metodice vykazovacího rámce ukončovat verze objektů, (např. Výkaz, Blok, Datová oblast, atd.), které nespadají do Knihovny.	Závazný	1
MVRA_6.0	Metodika vykazovacího rámce – zařazení verze Výkazu	Systém umožňuje uživateli zařazení nové verze Výkazu do právě jedné Metodiky vykazovacího rámce. Musí platit pravidla (viz kapitola 3.1.1 Metodika vykazovacího rámce).	Závazný	1
MVRA_7.0	Metodika	Systém přiřazuje verze objektů z Knihovny potřebných pro projektování	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	vykazovacího rámce –verze objektů Knihovny pro projektování	dle pravidel viz kapitola 3.1.1 Metodika vykazovacího rámce. V případě potřeby je možné přiřazenou verzi objektu ignorovat a vybrat verzi jinou (viz kapitola 3.1.1 Metodika vykazovacího rámce).		
MVRA_7.1	Metodika vykazovacího rámce – kontroly konzistence	V rámci Metodiky vykazovacího rámce probíhají kontroly konzistence (viz kapitola 2.5 Kontrola konzistence).	Závazný	1
MVRA_7.2	Metodika vykazovacího rámce – výsledky kontroly konzistence	Systém zobrazuje uživateli seznam objektů ve formě tabulky (gridu), u kterých kontrola konzistence skončila chybou. Umožňuje uživateli u Výkazů zobrazit seznam v nich použitých objektů ve formě tabulky (gridu), které danou chybu způsobily a u každého z nich vyznačí, o jakou chybu jde. Součástí výsledku kontroly konzistence je informace, zda pro odstranění chyby je případně nutné vytvořit novou verzi/variantu výkazu, nebo jiné řešení problému.	Závazný	1
MVRA_7.3	Metodika vykazovacího rámce – automatické verzování objektů	V rámci Metodiky vykazovacího rámce probíhá automatické verzování objektů (viz kapitola 2.3.3 Objekt Objekt_závislost).	Závazný	1
MVRA_8.0	Metodika vykazovacího rámce – přechod do Knihovny	Systém umožňuje uživateli přechod do části Knihovna, kde provede případné modifikace/založení verzí objektů Knihovny, a návrah zpět do původního místa v systému.	Závazný	1
MVRA_9.0	Metodika vykazovacího rámce – další	Metodika vykazovacího rámce umožňuje a zajišťuje další funkcionality např. vytvoření MVK mezi Metodikami vykazovacího rámce, přesouvat verze Výkazů mezi Metodikami vykazovacího rámce, nechat	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	funkce	rozpracovanou (a nezveřejněnou) verzi Výkazu v Metodice vykazovacího rámce apod. – více viz kapitola 3.1.1 Metodika vykazovacího rámce a procesy projektování.		
MVRA_10.0	Metodika vykazovacího rámce – ukončení platnosti Metodiky	Systém umožňuje ukončení platnosti Metodiky vykazovacího rámce v případě, že Metodika neobsahuje žádné verze Výkazů, jejichž platnost by byla delší než platnost ukončení Metodiky vykazovacího rámce. Po ukončení platnosti Metodiky vykazovacího rámce již nenásleduje další Metodika vykazovacího rámce daného Vykazovacího rámce.	Závazný	1

7.4 Výkaz ve Vykazovacím rámci

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
VYVRA_1.0	Výkaz ve Vykazovacím rámci – vznik vazby a atributy	Systém zajišťuje vznik vazby „Výkaz“ a „Vykazovací rámec“. Součástí vazby je atribut „zařazen dne“ (viz kapitola 3.1.2 Objekt Výkaz ve Vykazovacím rámci).	Závazný	1
VYVRA_2.0	Výkaz ve Vykazovacím rámci – změna zařazení Výkazu	Systém umožňuje uživateli změnit zařazení Výkazu do Vykazovacího rámce (viz kapitola 3.1.2 Objekt Výkaz ve Vykazovacím rámci).	Závazný	1
VYVRA_3.0	Výkaz ve Vykazovacím rámci – vliv na verze Výkazu v Metodikách	Systém hlídá vyřazení Výkazu z Vykazovacího rámce a zařazení do jiného Vykazovacího rámce dle kapitoly 3.1.2 Objekt Výkaz ve Vykazovacím rámci.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	vykazovacího rámce			

7.5 Verze výkazu v Metodice vykazovacího rámce

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
VVM_1.0	Verze výkazu v Metodice vykazovacího rámce – vznik vazby a atributy	Systém zajišťuje vznik vazby „Verze Výkaz“ a „Metodika Vykazovacího rámce“. Součástí vazby je atribut „enable“ (viz kapitola 3.1.3 Objekt Verze Výkazu v Metodice vykazovacího rámce).	Závazný	1
VVM_2.0	Verze výkazu v Metodice vykazovacího rámce – distribuce verze Výkazu	Systém umožňuje uživateli změnit nastavení atributu „enable“ za určitých podmínek (viz kapitola 3.1.3 Objekt Verze Výkazu v Metodice vykazovacího rámce), čímž ovlivní možnost distribuovat verzi Výkazu.	Závazný	1

7.6 Štítky výkazu

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
STI_1.0	Štítky Výkazu – vznik	Systém umožňuje uživateli založení objektu Štítek Výkazu k objektu Výkaz (viz kapitola 3.1.4 Objekt Štítky Výkazu).	Závazný	1
STI_2.0	Štítky Výkazu –	Systém nastaví platnost_od a platnost_do daného Štítku výkazu (viz	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	časová platnost	kapitola 3.1.4 Objekt Štítky Výkazu). Platnost_do může uživatel změnit.		

7.7 Výkaz

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
VYK_1.0	Výkaz - nový	Systém umožňuje uživateli založit první verzi instance objektu Výkaz (viz kapitola 5.3 Proces tvorby Výkazu), který je popsán standardními atributy (viz kapitola 3.2.1 Atributy objektu Výkaz) a neobsahuje vazby na žádné verze/varianty Bloku výkazu. První verze instance objektu Výkaz vzniká právě v jedné Metodice vykazovacího rámce a má systémem nastaven stav na Projektovaný.	Závazný	1
VYK_1.1	Výkaz - replikace	Systém umožňuje uživateli založit první verzi instance objektu Výkaz jako replikaci již existujícího Výkazu (viz kapitola 5.3 Proces tvorby Výkazu). Takto založený Výkaz obsahuje Bloky výkazu vytvořené podle BLV_1.1. Nová verze/varianta Výkazu vzniká právě v jedné Metodice vykazovacího rámce a má systémem nastaven stav na Projektovaný.	Závazný	1
VYK_1.3	Výkaz – sledování historie	Systém u každé instance objektu Výkaz sleduje jeho historii podle kapitoly 3.2 Objekt Výkaz.	Závazný	1
VYK_2.0	Výkaz – atributy nastavené systémem	Systém nastavuje Výkazu vytvořenému podle VYK_1.0 nebo VYK_1.1 následující atributy: • interní identifikátor objektu, • autor objektu (přihlášený uživatel),	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		• datum vytvoření (aktuální datum), • kdo aktualizoval (přihlášený uživatel), • datum a čas aktualizace (aktuální datum), • platnost_od (platnost_od Metodiky vykazovacího rámce), • platnost_do (platnost_do Metodiky vykazovacího rámce), • garant (viz OBE_8.0), • konzistence (ano), • synchronizovat na test (ne), • vlastník dat (viz VYK_2.6).		
VYK_2.1	Výkaz – atributy zadávané uživatelem	Systém umožňuje uživateli vyplnit atributy: • kód objektu, • název objektu, • popis objektu, • poznámka. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
VYK_2.2	Výkaz – atributy (nastavené systémem) měněné uživatelem	Systém umožňuje uživateli měnit následující atributy vyplněné podle VYK_2.0: • garant (výběrem ze seznamu zaměstnanců ČNB), • konzistence (boolean), • vlastník dat (viz VYK_2.6). Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
VYK_2.4	Výkaz – atributy měněné systémem	Systém nastavuje v případě změny některého atributu uživatelem podle VYK_2.1 a VYK_2.2 nebo vytvoření nové verze/varianty Výkazu nové	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		hodnoty atributům: • kdo aktualizoval (uživatel, který objekt vytvořil), • datum a čas aktualizace (aktuální datum).		
VYK_2.5	Výkaz – jednoznačnost atributu kód objektu	Systém zajišťuje unikátnost atributu kód objektu v rámci všech instancí objektu Výkaz. Nepovolí uživateli založit instanci objektu Výkaz s kódem objektu, který je již použit pro jinou instanci objektu Výkaz. Nepovolí uživateli změnit kód objektu u existující instance objektu Výkaz na hodnotu, která je použita pro jinou instanci objektu Výkaz. Kód objektu lze změnit jen v případě, že existuje pouze první verze, a to ve stavu Projektovaný, a není nikde použitá.	Závazný	1
VYK_2.6	Výkaz – atribut vlastník dat	Systém defaultně do atributu vlastník dat vyplní identifikaci organizačního útvaru ČNB, do kterého patří uživatel, jenž instanci objektu Výkaz vytvořil. Systém umožňuje uživateli změnit defaultní výběrem jiného identifikátoru organizačního útvaru ČNB ze seznamu.	Závazný	2
VYK_2.7	Výkaz – unikátnost atributu název objektu	Systém zajišťuje unikátnost názvu objektu pro všechny instance objektu Výkaz v rámci zvoleného časového řezu. Nepovolí uživateli založit v daném časovém řezu instanci objektu Výkaz s názvem objektu, který je již použit pro jinou instanci objektu Výkaz toho samého časového řezu. Nepovolí uživateli změnit název objektu u existující instance objektu Výkazu na hodnotu, která je v daném časovém řezu použita pro jinou instanci objektu Výkaz.	Závazný	1
VYK_3.0	Výkaz – vazba na Blok výkazu	Systém umožňuje uživateli do Výkazu zařadit neomezený počet Bloků výkazu (viz kapitola 3.3 Objekt Blok výkazu).	Závazný	1
VYK_3.1	Výkaz – vazba na Vykazovací rámec	Systém umožňuje uživateli zařadit Výkaz do neomezeného počtu Vykazovacích rámců. V jeden okamžik však může být Výkaz zařazen do právě jednoho Vykazovacího rámce.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
VYK_3.2	Výkaz – vazba na Kontrolu	Systém umožňuje uživateli na Výkaz navázat neomezeně instancí objektu Kontrola (viz kapitola 3.19 Objekt Kontrola).	Závazný	1
VYK_4.0	Výkaz – změna atributů	Systém umožňuje uživateli měnit atributy verze/varianty Výkazu v souladu se stanovenými pravidly (viz kapitola 2.3 Vazby mezi jednotlivými objekty).	Závazný	1
VYK_4.1	Výkaz – změna vazeb na podřízené objekty	Systém umožňuje uživateli změnit vazby verze/varianty Výkazu na jemu podřízené objekty pouze pokud se verze/varianta Výkazu nachází ve stavu Projektovaný.	Závazný	1
VYK_4.2	Výkaz – schválení	Systém umožňuje uživateli schválit instanci objektu Výkaz (viz OBE_4.0). Výkaz bez alespoň jednoho Bloku výkazu nelze schválit.	Závazný	1
VYK_4.3	Výkaz – schválení s klonovanou Datovou oblasti	Systém umožňuje uživateli schválit instanci objektu Výkaz, který obsahuje klonovanou Datovou oblast (viz DOB_1.2) pouze po splnění podmínek uvedených ve VYK_4.2 a podmínky, že instance objektu Výkaz, který obsahuje mateřskou Datovou oblast, je ve stavu Schválený nebo Platný.	Závazný	1
VYK_4.4	Výkaz – změna stavu ze stavu Schválený na stav Projektovaný	Systém umožňuje uživateli změnit stav instance objektu Výkaz ze stavu Schválený na stav Projektovaný (viz OBE_6.0).	Závazný	1
VYK_4.5	Výkaz – zplatnění	Systém umožňuje uživateli zplatnit instanci objektu Výkaz (viz OBE_5.0).	Závazný	1
VYK_4.6	Výkaz – změna varianty na verzi	Systém umožňuje uživateli změnit vytvořenou variantu instance objektu Výkaz na verzi instance objektu Výkaz (viz OBE_7.0).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
VYK_5.0	Výkaz – ukončení platnosti uživatelem	Systém umožňuje uživateli ukončit platnost instanci objektu Výkaz (viz OBE_2.0).	Závazný	1
VYK_5.1	Výkaz – ukončení platnosti systémem v závislosti na zplanění jiné verze/varianty téhož objektu	Systém automaticky ukončuje platnost instanci objektu Výkaz (viz OBE_2.1).	Závazný	1
VYK_5.2	Výkaz – prodloužení platnosti uživatelem	Systém umožňuje uživateli prodloužit platnost instance objektu Výkaz (viz OBE_3.0).	Závazný	1
VYK_6.0	Výkaz – smazání	Systém umožňuje uživateli smazat instanci objektu Výkaz (viz OBE_1.0).	Závazný	1
VYK_7.0	Výkaz – zobrazení seznamu	Systém umožňuje uživateli zobrazit seznam obsahující verze/varianty Výkazů ve formě tabulky (grid).	Závazný	1
VYK_7.1	Výkaz – vytvoření struktury systémem	Systém vytváří strukturu verze/varianty Výkazu na základě struktury jemu podřízených instancí objektů Blok výkazu a Datová oblast. Struktura těchto objektů je určena objekty určujícími dimenze Datových oblastí (viz kapitola 3.4 Objekt Datová oblast).	Závazný	1
VYK_7.4	Výkaz – zobrazení struktury	Systém umožňuje uživateli zobrazit strukturu verze/varianty Výkazu (viz kapitola 5.3 Proces tvorby Výkazu) výběrem ze seznamu (viz VYK_7.0).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
VYK_7.5	Výkaz – export struktury	Systém umožňuje uživateli zobrazenou strukturu verze/varianty Výkazu (viz VYK_7.4) exportovat do formátů DOC, DOCX, XLS, XLSX.	Závazný	2
VYK_7.6	Výkaz – prezentační vrstva	Systém umožňuje uživateli zobrazit prezentační vrstvu verze/varianty Výkazu (viz kapitola 5.3 Proces tvorby Výkazu) výběrem ze seznamu (viz VYK_7.0).	Závazný	2
VYK_7.7	Výkaz – úprava prezentační vrstvy	Systém umožňuje uživateli upravit prezentační vrstvu verze/varianty Výkazu v souladu s kapitolou 5.3 Proces tvorby Výkazu)	Závazný	2
VYK_7.8	Výkaz – export prezentační vrstvy	Systém umožňuje uživateli zobrazenou prezentační vrstvu verze/varianty Výkazu (viz VYK_7.6) exportovat do formátů DOC, DOCX, XLS, XLSX.	Závazný	2
VYK_8.0	Výkaz – prezentace internímu uživateli	Systém prezentuje instance objektu Výkaz interním uživatelům v plném rozsahu a historii (viz kapitola 5.7 Proces prezentace Výkazu). Interním uživatelům jsou prezentovány Výkazy ve všech stavech (Projektovaný, Schválený i Platný).	Závazný	1
VYK_8.1	Výkaz – prezentace Osobám	Systém prezentuje instance objektu Výkaz Osobám v rozsahu a historii určené interním uživatelem (viz kapitola 5.7 Proces prezentace Výkazu). Instance objektu Výkaz jsou Osobám prezentovány ve formátech XML, XBRL (reference na taxonomii vytvořenou mimo SDAT), HTML v závislosti na nastavení možných formátů na úrovni instance objektu Výkaz a zvolené technologii pro sběr dat (viz dokument D – Sběr dat, kapitola 5 Formáty pro výměnu dat) Systém v produkčním prostředí Osobám prezentuje instance objektu Výkaz, které jsou ve stavech Schválený a Platný.	Závazný	2
VYK_8.2	Výkaz –	Systém umožňuje prezentovat Osobám instance objektu Výkaz, které jsou	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	předběžná prezentace Osobám	ve stavu Projektovaný prostřednictvím testovacího prostředí.		
VYK_8.3	Výkaz – export do testovacího prostředí	Systém umožňuje uživateli exportovat instance objektu Výkaz (včetně všech na něj navázaných objektů), které jsou ve stavu Projektovaný, Schválený nebo Platný, do testovacího prostředí (viz dokument A – Obecné požadavky, kapitola 2.1.2 Testovací prostředí).	Závazný	2
VYK_8.4	Výkaz – prezentace provedených změn	Systém umožňuje uživateli vybrat k prezentaci pouze změny provedené v poslední verzi/variantě Výkazu oproti přechozí verzi/variantě Výkazu.	Závazný	2
VYK_9.0	Výkaz – kontrola konzistence	Systém umožňuje uživateli spustit kontrolu konzistence (viz kapitola 2.5 Kontrola konzistence) nad vybranou instancí objektu Výkaz.	Závazný	1
VYK_9.1	Výkaz – výsledek kontroly konzistence	Systém zobrazuje uživateli seznam objektů, ve formě tabulky (gridu), které způsobily chybu kontroly konzistence nad vybraným výkazem a u každého z nich vyznačí, o jakou chybu jde. Součástí výsledku kontroly konzistence je informace, zda pro odstranění chyby je nutné vytvořit novou verzi/variantu výkazu.	Závazný	1
VYK_9.2	Výkaz – uživatelský zámek	Systém zamyká Výkaz (či jeho části). V okamžiku, kdy uživatel začne editovat Výkaz nebo jeho část, je tento Výkaz nebo jeho část uzamčena a není možné, aby editaci Výkazu nebo jeho části po dobu trvání zámku prováděl jiný uživatel, i když k tomu má oprávnění. Toto opatření má zabránit náhodnému a nechtěnému zásahu do již rozdělené práce. Zámek nad Výkazem nebo jeho části bude uvolněn v okamžiku, kdy uživatel, který zámek vytvořil, rozhodne o jeho uvolnění. Možností pro uvolnění zámku je odchod z editačního formuláře (ovládacím prvkem typu opustit formulář apod.) kdy je uživatel dotázán, zda si přeje zámek ponechat.	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		V případě, že by nastala situace, kdy je potřeba uvolnit zámek v případě, že jej autor zámku z nějakých důvodů nemůže uvolnit, provede toto administrátor systému na žádost uživatele, který má právo Výkaz editovat.		

7.8 Blok výkazu

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
BLV_1.0	Blok výkazu - nový	Systém umožňuje uživateli založit první verzi instance objektu Blok výkazu (viz kapitola 5.3 Proces tvorby Výkazu).který je popsán standardními atributy (viz kapitola 3.3.1 Atributy objektu Blok výkazu) a neobsahuje vazby na žádné verze varianty Datové oblasti. První verze instance objektu Blok výkazu vzniká právě v jedné instanci objektu Výkaz, jenž je ve stavu Projektovaný. Tato verze Bloku výkazu má systémem nastaven stav na Projektovaný.	Závazný	1
BLV_1.1	Blok výkazu – replikace	Systém umožňuje uživateli založit Blok výkazu jako replikaci již existujícího Bloku výkazu (viz kapitola 5.3 Proces tvorby Výkazu). Takto založený Blok výkazu obsahuje Datové oblasti vytvořené podle DOB_1.1. Nová verze/varianta Bloku výkazu vzniká právě v jedné instanci objektu Výkaz, jenž je ve stavu Projektovaný. Tato verze/varianta Bloku výkazu má systémem nastaven stav na Projektovaný.	Závazný	1
BLV_1.2	Blok výkazu – vytvoření verze/varianty	Systém umožňuje uživateli vytvořit verzi/variantu jakékoliv instanci objektu Blok výkazu, jenž je v instanci objektu Výkaz, který je ve stavu Projektovaný.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	uživatelem			
BLV_1.3	Blok výkazu – vytvoření verze/varianty systémem	Systém vytváří novou verzi/variantu Bloku výkazu, pouze pokud je v instanci objektu Výkaz, který je ve stavu Projektovaný. Systém vytváří novou verzi/variantu instance objektu Blok výkazu (viz kapitola 2.3 Vazby mezi jednotlivými objekty), pokud byla uživatelem nebo systémem vytvořena nová verze/varianta jemu podřízeného objektu Datová oblast.	Závazný	1
BLV_1.4	Blok výkazu – sledování historie	Systém u každé instance objektu Blok výkazu sleduje jeho historii podle kapitoly 3.3 Objekt Blok výkazu)	Závazný	1
BLV_2.0	Blok výkazu – atributy nastavené systémem	Systém nastavuje Bloku výkazu vytvořenému podle BLV_1.0 nebo BLV_1.1 následující atributy: • interní identifikátor objektu, • autor objektu (přihlášený uživatel), • datum vytvoření (aktuální datum), • kdo aktualizoval (přihlášený uživatel), • datum a čas aktualizace (aktuální datum), • platnost_od (platnost_od výkazu), • platnost_do (platnost_od výkazu), • pořadí (dle pořadí jeho vytvoření v instanci objektu Výkaz).	Závazný	1
BLV_2.1	Blok výkazu – atributy zadávané uživatelem	Systém umožňuje uživateli vyplnit atributy: • kód objektu, • název objektu, • popis objektu, • poznámka. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.		
BLV_2.2	Blok výkazu – atributy měněné systémem	Systém nastavuje v případě změny některého atributu uživatelem podle BLV_2.1 nebo vytvoření nové verze/varianty Bloku výkazu nové hodnoty atributům: • kdo aktualizoval (uživatel, který objekt vytvořil), • datum a čas aktualizace (aktuální datum).	Závazný	1
BLV_2.3	Blok výkazu – unikátnost atribut název objektu	Systém zajišťuje unikátnost atributu název objektu Bloku výkazu v rámci jedné verze/varianty instance objektu Výkaz. Nepovolí uživateli založit Blok výkazu s názvem objektu, který je již použit pro jiný Blok výkazu v rámci jedné verze/varianty instance objektu Výkaz. Nepovolí uživateli změnit název objektu u existujícího Bloku výkazu na hodnotu, která je použita pro jiný Blok výkazu v rámci jedné verze/varianty instance objektu Výkaz.	Závazný	1
BLV_2.4	Blok výkazu – atributy (nastavené systémem) měněné uživatelem	Systém umožňuje uživateli změnit atribut pořadí nastavený podle BLV_2.0 za dodržení podmínek stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
BLV_2.5	Blok výkazu – jednoznačnost kódu objektu	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Blok výkazu je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Blok Výkazu. Kód objektu lze změnit jen v případě, že existuje pouze první verze, a to ve stavu Projektovaný.	Závazný	1
BLV_3.0	Blok výkazu – vazba na Datovou	Systém umožňuje uživateli do Bloku výkazu zařadit neomezený počet Datových oblastí (viz kapitola 3.3 Objekt Blok výkazu).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	oblast			
BLV_4.0	Blok výkazu – změna atributů	Systém umožňuje uživateli měnit atributy verze/varianty Bloku výkazu v souladu se stanovenými pravidly v kapitole 2.3 Vazby mezi jednotlivými objekty).	Závazný	1
BLV_4.1	Blok výkazu – změna vazeb na podřízené a nadřízené objekty	Systém umožňuje uživateli změnit vazby verze/varianty Bloku výkazu na jemu podřízené a nadřízené objekty, pouze pokud se verze/varianta Bloku výkazu a jemu nadřízená verze/varianta Výkazu nacházejí ve stavu Projektovaný.	Závazný	1
BLV_4.2	Blok výkaz – schválení	Systém schvaluje instanci objektu Blok výkazu v rámci schvalování jemu nadřízené instance objektu Výkaz (viz VYK_4.2). Blok výkazu bez alespoň jedné Datové oblasti nelze schválit.	Závazný	1
BLV_4.4	Blok výkazu – zplatnění	Systém zplatňuje instanci objektu Blok výkazu v rámci zplatňování jemu nadřízené instance objektu Výkaz (viz VYK_4.2).	Závazný	1
BLV_4.5	Blok výkazu – změna varianty na verzi	Systém umožňuje uživateli změnit vytvořenou variantu instance objektu Blok výkazu na verzi instance objektu Blok výkazu (viz OBE_7.0).	Závazný	1
BLV_5.0	Blok výkazu – ukončení platnosti uživatelem	Systém umožňuje uživateli ukončit platnost instanci objektu Blok výkazu (viz OBE_2.0).	Závazný	1
BLV_5.1	Blok výkazu – ukončení platnosti systémem v závislosti na zplanění jiné verze/varianty	Systém automaticky ukončí platnost instanci objektu Blok výkazu (viz OBE_2.1).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	téhož objektu			
BLV_5.2	Blok výkazu – prodloužení platnosti	Systém prodlužuje platnost instance objektu Blok výkazu v rámci prodloužení platnosti nadřízené instance objektu Výkaz (viz VYK_5.2).	Závazný	1
BLV_6.0	Blok výkazu – smazání	Systém umožňuje uživateli smazat instanci objektu Blok výkazu (viz OBE_1.0).	Závazný	1
BLV_7.0	Blok výkazu – zobrazení seznamu všech	Systém umožňuje uživateli zobrazit seznam obsahující všechny verze/varianty Bloků výkazů ve formě tabulky (grid).	Závazný	1
BLV_7.1	Blok výkazu – zobrazení seznamu podle Výkazu	Systém umožňuje uživateli zobrazit seznam obsahující všechny verze/varianty Bloků výkazů jedné instance objektu Výkaz ve formě tabulky (grid).	Závazný	1
BLV_7.2	Blok výkazu – vytvoření struktury systémem	Systém vytváří strukturu verze/varianty Bloku výkazu na základě struktury jemu podřízených instancí objektů Datová oblast. Struktura těchto objektů je určena objekty určujícími dimenze Datových oblastí (viz kapitola 3.4 Objekt Datová oblast).	Závazný	1
BLV_7.5	Blok výkazu – zobrazení struktury	Systém umožňuje uživateli zobrazit strukturu verze/varianty Bloku výkazu (viz kapitola 5.3 Proces tvorby Výkazu) výběrem ze seznamu (viz BLV_7.0 a BLV_7.1).	Závazný	1
BLV_7.6	Blok výkazu – export struktury	Systém umožňuje uživateli zobrazenou strukturu verze/varianty Bloku výkazu (viz BLV_7.5) exportovat do formátů DOC, DOCX, XLS, XLSX.	Závazný	2
BLV_7.7	Blok výkazu – zobrazení prezentační vrstvy	Systém umožňuje uživateli zobrazit prezentační vrstvu verze/varianty Bloku výkazu (viz kapitola 5.3 Proces tvorby Výkazu) výběrem ze seznamu (viz BLV_7.0 nebo BLV_7.1).	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
BLV_7.8	Blok výkazu – úprava prezentační vrstvy	Systém umožňuje uživateli upravit prezentační vrstvu verze/varianty Bloku výkazu v souladu s kapitolou 5.3 Proces tvorby Výkazu.	Závazný	2
BLV_7.9	Blok výkazu – export prezentační vrstvy	Systém umožňuje uživateli zobrazenou prezentační vrstvu verze/varianty Výkazu (viz BLV_7.7) exportovat do formátů DOC, DOCX, XLS, XLSX.	Závazný	2
BLV_8.0	Blok výkazu – prezentace internímu uživateli	Systém prezentuje instance objektu Blok výkazu interním uživatelům v plném rozsahu a historii (viz kapitola 5.7 Proces prezentace Výkazu). Tato prezentace je možná v rámci kontextu celé instance objektu Výkaz nebo instance objektu Blok výkazu samostatně. Interním uživatelům jsou prezentovány Bloky výkazu ve všech stavech (Projektovaný, Schválený i Platný).	Závazný	1
BLV_8.1	Blok výkazu – prezentace Osobám	Systém prezentuje instance objektu Blok výkazu Osobám pouze v rámci prezentace instance objektu Výkaz (viz VYK_8.1).	Závazný	2

7.9 Datová oblast

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DOB_1.0	Datová oblast - nová	Systém umožňuje uživateli založit první verzi instance objektu Datová oblast (viz kapitola 5.3 Proces tvorby Výkazu), který je popsán standardními atributy (viz kapitola 3.4.1 Atributy objektu Datová oblast) a neobsahuje vazby na žádné verze/varianty objektů popisujících údaje. První verze instance objektu Datová oblast vzniká právě v jedné instanci	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		objektu Blok výkazu, jenž je ve stavu Projektovaný. Tato verze Datové oblasti má systémem nastaven stav na Projektovaný.		
DOB_1.1	Datová oblast – replikace	Systém umožňuje uživateli založit Datovou oblast jako replikaci již existující Datové oblasti (viz kapitola 5.3 Proces tvorby Výkazu). Takto založená instance objektu Datová oblast obsahuje vazby na verze/varianty objektů popisujících Údaje. Nová verze/varianta Datové oblasti vzniká právě v jedné instanci objektu Blok výkazu, jenž je ve stavu Projektovaný. Tato verze/varianta Datové oblasti má systémem nastaven stav na Projektovaný.	Závazný	1
DOB_1.2	Datová oblast – klonování	Systém umožňuje uživateli založit Datovou oblast klonováním již existující instance objektu Datová oblasti (viz kapitola 3.4 Objekt Datová oblast). Klonováním je systémem nové Datové oblasti (potomek mateřské Datové oblasti) do atributu „mateřská Datová oblast“ vyplněn interní identifikátor instance objektu Datová oblast, z které klon vznikl. Takto založená instance objektu Datová oblast obsahuje vazby na stejné verze/varianty objektů popisujících Údaje jako mateřská Datová oblast. Systém umožňuje klonovat jakoukoliv instanci objektu Datová oblast, která nevznikla v důsledku klonování (tj. klonovanou Datovou oblast nelze klonovat) a instance objektu Výkaz, v níž je zařazena, je ve stavu Schválený nebo Platný. Instance objektu Datová oblast, která je zařazena v instanci objektu Výkaz, jenž je ve stavu Projektovaný, může být klonována pouze v případě, že má instance objektu Datová oblast atribut „umožnit sdílení před schválením výkazu“ nastaven na hodnotu „ano“.	Závazný	2
DOB_1.3	Datová oblast – vytvoření	Systém umožňuje uživateli vytvořit verzi/variantu jakékoliv instanci objektu Datová oblast, jenž je v instanci objektu Blok výkazu, který je ve	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	verze/varianty uživatelem	stavu Projektovaný.		
DOB_1.4	Datová oblast – vytvoření verze/varianty systémem	Systém vytváří novou verzi/variantu instance objektu Datová oblast (viz kapitola 2.3 Vazby mezi jednotlivými objekty), pokud byla uživatelem nebo systémem vytvořena nová verze/varianta instancí objektů popisujících Údaje, které jsou v Datové oblasti použity (viz kapitola 5.4 Proces tvorby objektů popisujících údaje). Systém vytváří novou verzi/variantu Datové oblasti, pouze pokud je v instanci objektu Výkaz, který je ve stavu Projektovaný.	Závazný	1
DOB_1.5	Datová oblast – sledování historie	Systém u každé instance objektu Datová oblast sleduje její historii podle kapitoly 3.4 Objekt Datová oblast).	Závazný	1
DOB_2.0	Datová oblast – atributy nastavené systémem	Systém nastavuje Datové oblasti vytvořené podle DOB_1.0, DOB_1.1 nebo DOB_1.2 následující atributy: • interní identifikátor objektu, • autor objektu (přihlášený uživatel), • datum vytvoření (aktuální datum), • kdo aktualizoval (přihlášený uživatel), • datum a čas aktualizace (aktuální datum), • platnost_od (platnost_od bloku výkazu), • platnost_do (platnost_do bloku výkazu), • pořadí (dle pořadí jejího vytvoření v rámci instance objektu Blok výkazu), • garant (viz OBE_8.0), • umožnit sdílení před schválením výkazu (ne).	Závazný	1
DOB_2.1	Datová oblast – atributy zadávané	Systém umožňuje uživateli vyplnit atributy:	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	uživatelem	• kód objektu, • název objektu, • popis objektu, • poznámka, • typ datové oblasti. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.		
DOB_2.2	Datová oblast – atributy měněné systémem	Systém nastavuje v případě změny některého atributu uživatelem podle DOB_2.1 nebo vytvoření nové verze/varianty Datové oblasti nové hodnoty atributům: • kdo aktualizoval (uživatel, který objekt vytvořil), • datum a čas aktualizace (aktuální datum).	Závazný	1
DOB_2.3	Datová oblast – atributy (zadané systémem) měněné uživatelem	Systém umožňuje uživateli měnit atributy zadané podle DOB_2.0: • garant objektu (výběrem ze seznamu zaměstnanců ČNB), • umožnit sdílení před schválením výkazu, • pořadí, které jsou nastaveny systémem, za splnění podmínek uvedených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
DOB_2.4	Datová oblast – unikátnost atributu název objektu	Systém zajišťuje unikátnost atributu název objektu instance objektu Datová oblast v rámci jedné verze/varianty instance objektu Výkaz. Nepovolí uživateli založit Datovou oblast s názvem objektu, který je již použit pro jinou Datovou oblast v rámci jedné verze/varianty instance objektu Výkaz. Nepovolí uživateli změnit název objektu u existující Datové oblasti na hodnotu, která je použita pro jinou Datovou oblasti v rámci jedné verze/varianty instance objektu Výkaz.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DOB_2.5	Datová oblast – jednoznačnost atributu kód objektu	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Datová oblast je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Datová oblast. Kód objektu lze změnit jen v případě, že existuje pouze první verze, a to ve stavu Projektovaný.	Závazný	1
DOB_3.0	Datová oblast – vazba na Údaj	Systém umožňuje uživateli v Datové oblasti vytvořit neomezený počet Údajů (viz kapitola 3.4 Objekt Datová oblast).	Závazný	1
DOB_4.0	Datová oblast – změna atributů	Systém umožňuje uživateli měnit atributy verze/varianty Datové oblasti v souladu s pravidly stanovenými v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
DOB_4.1	Datová oblast – změna vazeb na podřízené a nadřízené objekty	Systém umožňuje uživateli změnit vazby verze/varianty Datové oblasti na jemu podřízené a nadřízené objekty, pouze pokud se verze/varianta Datové oblasti a jemu nadřízená verze/varianta Bloku výkazu nacházejí ve stavu Projektovaný.	Závazný	1
DOB_4.2	Datová oblast – použití objektů popisujících Údaje	Systém umožňuje uživateli v instanci objektu Datová oblast použít neomezený počet objektů popisujících Údaje (viz kapitola 3.18 Objekt Údaj) podle pravidel definovaných v (viz kapitola 5.3 Proces tvorby Výkazu). Systém umožňuje uživateli vybírat instance objektů popisujících Údaje ze seznamu ve formě tabulky (grid), který je přístupný z okruhu Knihovna (viz kapitola 5.3 Proces tvorby Výkazu).	Závazný	1
DOB_4.3	Datová oblast – hromadné použití objektů popisujících Údaje	Systém umožňuje uživateli použít instance objektů popisujících Údaje pro více Údajů v rámci jedné Datové oblasti najednou (viz kapitola 5.3.2.1 Formulářové prostředí).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DOB_4.4	Datová oblast – informace o nové verzi/variantě mateřské Datové oblasti	Systém informuje uživatele o vytvoření nové verze/varianty mateřské Datové oblasti (viz kapitola 5.3 Proces tvorby Výkazu).	Závazný	2
DOB_4.5	Datová oblast – informace o zaverzování Výkazu obsahujícího klonovanou Datovou oblast	Systém informuje uživatele, že došlo k zaverzování instance objektu Výkaz, jenž obsahuje klonovanou Datovou oblast (viz kapitola 5.3 Proces tvorby Výkazu).	Závazný	2
DOB_4.6	Datová oblast – propagace změn v mateřské Datové oblasti do klonů	Systém automaticky propaguje změny provedené ve verzi/variantě mateřské Datové oblasti do všech jejích klonů, které jsou součástí instancí objektu Výkaz, jež jsou zařazeny v projektované Metodice vykazovacího rámce.	Závazný	2
DOB_4.7	Datová oblast – zrušení vazby mateřské Datové oblasti a klonované Datové oblasti	Systém zruší všechny vazby modifikované mateřské Datové oblasti na všechny její klony, jež jsou součástí instancí objektů Výkaz, které nejsou zařazeny do žádné projektované Metodiky vykazovacího rámce. Systém zruší všechny vazby klonovaných Datových oblastí na mateřskou Datovou oblast v případě, že uživatel modifikuje klonovanou Datovou oblast (viz kapitola 5.3 Proces tvorby Výkazu).	Závazný	2
DOB_4.8	Datová oblast – schválení	Systém schvaluje instanci objektu Datová oblast v rámci schvalování jemu nadřízené instance objektu Blok výkazu (viz BLV_4.2) Datovou oblast bez alespoň jednoho Údaje nelze schválit.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DOB_4.9	Datová oblast – zrušení vazby klonované Datové oblasti a mateřské Datové oblasti uživatelem	Systém umožňuje uživateli zrušit vazbu vybrané klonované Datové oblasti na její mateřskou Datovou oblast.	Závazný	2
DOB_4.10	Datová oblast – zplatnění	Systém zplatňuje instanci objektu Datová oblast v rámci zplatňování jemu nadřízené instance objektu Blok výkazu (viz BLV_4.4).	Závazný	1
DOB_4.11	Datová oblast – potlačení vykazování	Systém umožňuje uživateli v rámci Datové oblasti potlačit vykazování pro vybraný Údaj, čímž je v rámci instance objektu Datová oblast zamezeno vzniku Údaje v databázi.	Závazný	1
DOB_4.12	Datová oblast – vyhledávání duplicit Údajů	Systém informuje uživatele o vzniku více Údajů se stejnou konkretizací (viz kapitola 3.18 Objekt Údaj) v rámci Datových oblastí, které jsou součástí jedné instance objektu Výkaz (viz kapitola 5.3 Proces tvorby Výkazu).	Závazný	1
DOB_4.13	Datová oblast – změna varianty na verzi	Systém umožňuje uživateli změnit vytvořenou variantu instance objektu Datová oblast ve stavu Projektovaný na verzi instance objektu Datová oblast (viz OBE_7.0).	Závazný	1
DOB_4.14	Datová oblast – zobrazení údaje na více místech výkazu	Systém umožňuje uživateli určit u duplicitních údajů (viz DOB_4.12) právě jeden z nich, který bude vykazovaný. Systém označí automaticky zbylé duplicitní údaje jako zobrazované.	Závazný	1
DOB_5.0	Datová oblast – ukončení platnosti uživatelem	Systém umožňuje uživateli ukončit platnost instanci objektu Datová oblast (viz OBE_2.0).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DOB_5.1	Datová oblast – ukončení platnosti systémem v závislosti na zplanění jiné verze/varianty téhož objektu	Systém automaticky ukončuje platnost instanci objektu Datová oblast (viz OBE_2.1).	Závazný	1
DOB_5.2	Datová oblast – prodloužení platnosti	Systém prodlužuje platnost instance objektu Datová oblast v rámci prodloužení platnosti nadřízené instance objektu Blok výkazu (viz BLV_5.2).	Závazný	1
DOB_6.0	Datová oblast – smazání	Systém umožňuje uživateli smazat instanci objektu Datová oblast (viz OBE_1.0).	Závazný	1
DOB_7.0	Datová oblast – zobrazení seznamu všech Datových oblastí	Systém umožňuje uživateli zobrazit seznam obsahující všechny verze/varianty Datových oblastí ve formě tabulky (grid).	Závazný	1
DOB_7.1	Datová oblast – zobrazení seznamu podle Výkazů	Systém umožňuje uživateli zobrazit seznam obsahující všechny verze/varianty Datových oblastí jedné instance objektu Výkaz ve formě tabulky (grid).	Závazný	1
DOB_7.2	Datová oblast – zobrazení seznamu podle Bloku výkazu	Systém umožňuje uživateli zobrazit seznam obsahující všechny verze/varianty Datových oblastí jedné instance objektu Blok výkazu ve formě tabulky (grid).	Závazný	1
DOB_7.3	Datová oblast – vytvoření	Systém vytváří strukturu verze/varianty Datové oblasti na základě použití objektů popisujících Údaje Datové oblasti (viz kapitola 5.3.2.1	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	struktury systémem	Formulářové prostředí). Struktura této Datové oblasti je určena objekty určujícími dimenze Datových oblastí (viz kapitola 3.4 Objekt Datová oblast).		
DOB_7.6	Datová oblast – zobrazení struktury	Systém umožňuje uživateli zobrazit strukturu verze/varianty Datové oblasti (viz kapitola 5.3 Proces tvorby Výkazu) výběrem ze seznamu (viz DOB_7.0, DOB_7.1 a DOB_7.2).	Závazný	1
DOB_7.7	Datová oblast – export struktury	Systém umožňuje uživateli zobrazenou strukturu verze/varianty Datové oblasti (viz BLV_7.6) exportovat do formátů DOC, DOCX, XLS, XLSX.	Závazný	2
DOB_7.8	Datová oblast – zobrazení prezentační vrstvy	Systém umožňuje uživateli zobrazit prezentační vrstvu verze/varianty Datové oblasti (viz kapitola 5.3 Proces tvorby Výkazu) výběrem ze seznamu (viz DOB_7.0, DOB_7.1 a DOB_7.2).	Závazný	2
DOB_7.9	Datová oblast – úprava prezentační vrstvy	Systém umožňuje uživateli upravit prezentační vrstvu verze/varianty Datové oblasti v souladu s (viz kapitola 5.3 Proces tvorby Výkazu).	Závazný	2
DOB_7.10	Datová oblast – export prezentační vrstvy	Systém umožňuje uživateli zobrazenou prezentační vrstvu verze/varianty Datové oblasti (viz DOB_7.8) exportovat do formátů DOC, DOCX, XLS, XLSX.	Závazný	2
DOB_8.0	Datová oblast – prezentace internímu uživateli	Systém prezentuje instance objektu Datová oblast interním uživatelům v plném rozsahu a historii (viz kapitola 5.7 Proces prezentace Výkazu). Tato prezentace je možná v rámci kontextu celé instance objektu Výkaz nebo instance objektu Blok výkazu nebo instance objektu Datová oblast samostatně. Interním uživatelům jsou prezentovány Datové oblasti ve všech stavech (Projektovaný, Schválený i Platný).	Závazný	1
DOB_8.1	Datová oblast –	Systém prezentuje instance objektu Datová oblast Osobám pouze v rámci	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	prezentace Osobám	prezentace instance objektu Výkaz (viz VYK_8.1).		
DOB_11.0	Identifikační Parametr v Datové oblasti	Systém umožňuje uživateli použít na dynamické Datové oblasti Parametr typu identifikační.	Závazný	1
DOB_11.1	Nastavení kontrol jednoznačnosti v dynamické Datové oblasti	Systém umožňuje uživateli u dynamické Datové oblasti, která obsahuje identifikační Parametr, nastavit způsob kontroly jednoznačnosti dynamických řádků v Datové oblasti při provádění formátových kontrol: a) kontrola jednoznačnosti pouze hodnoty identifikačního Parametru nebo b) kontrola jednoznačnosti hodnot identifikačního Parametru a zároveň jednoznačnosti kombinace hodnot zbývajících dynamických Parametrů (default).	Závazný	1

7.10 Číselník, Položka číselníku

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
CIS_1.0	Vytvoření prázdné instance Číselník	Systém umožňuje vytvoření prázdné instance objektu Číselník podle byznys pravidel definovaných v kapitole 3.5 Objekt Číselník. Při vytvoření nové instance objektu Číselník systém automaticky založí novou verzi Číselníku a automaticky tuto verzi založí ve stavu Projektovaný. Podrobně je popis životního cyklu instancí popsán v kapitole 2.2 Sledování historie objektů.	Závazný	1
CIS_1.1	Jednoznačnost kódu objektu	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Číselník je jednoznačná v porovnání s hodnotami téhož	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	Číselník	atributu v rámci již existujících instancí objektu Číselník. Systém nepovoluje změnit kód objektu na hodnotu již použitou u jiné instance objektu Číselník.		
CIS_1.2.	Formát kódu Číselníku	Kód instance objektu Číselník se skládá z prefixu a pořadového čísla: • prefix vybírá uživatel nabídkou z listu číselníkových položek, např. BA, SE, viz CIS_1.4, k vybranému prefixu systém nabízí číselnou 4-místnou hodnotu, tj. například SE0001 pro první Číselník odvozený od prefixu SE. Pro další Číselníky systém nabízí číselnou hodnotu vyšší o 1 (např. SE0002).	Závazný	1
CIS_1.3	Změna kódu objektu Číselník	Systém umožňuje uživateli změnit kód instance objektu Číselník v případě, že existuje pouze první verze, a to ve stavu Projektovaný a není nikde použita. Uživatel kód instance objektu změní jeho přepsáním.	Závazný	1
CIS_1.4	Prefix kódu objektu Číselník	Systém umožňuje uživateli aktualizovat (zakládat nové, ukončovat platnost a mazat nepoužité položky) číselník prefixů kódů pro objekt Číselník.	Závazný	1
CIS_2.0	Editace objektu Číselník	Systém umožňuje měnit atributy existujícího objektu Číselník za splnění byznys podmínek definovaných v kapitole 3.5 Objekt Číselník. Uživatel má možnost se rozhodnout zda vytvoří novou verzi nebo variantu objektu. Typy změn, a zdali je možné provést ve verzi či variantě pro objekt Číselník, musí být v souladu s nastavením systému. Podrobně jsou pravidla uvedena v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
CIS_3.0	Smazání objektu Číselník	Systém umožňuje smazat objekt Číselník, viz OBE_1.0. Smazání objektu instance Číselník je povoleno za následujících podmínek:	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		• z Číselníku není vytvořena žádná Hierarchie číselníku, • z Číselníku není vytvořena žádná Doména číselníku, • není použita žádná Položka číselníku pro dodatečnou konkretizaci Ukazatele.		
CIS_4.0	Ukončení platnosti objektu Číselník	Systém umožňuje ukončit platnost instanci objektu (viz OBE_2.0 a OBE_2.1).	Závazný	1
CIS_5.0	Prodloužení platnosti instance objektu Číselník	Systém umožňuje instanci objektu Číselník prodloužit platnost (viz OBE_3.0).	Závazný	1
CIS_6.0	Schválení instance objektu Číselník	Systém umožňuje schválit instanci objektu Číselník (viz OBE_4.0).	Závazný	1
CIS_7.0	Zplatnění instance objektu Číselník	Systém umožňuje zplatnit instanci objektu Číselník (viz OBE_5.0).	Závazný	1
CIS_8.0	Změna stavu instance objektu Číselník ze stavu Schválený na Projektovaný	Systém umožňuje změnu stavu instance objektu Číselník ze stavu Schválený do stavu Projektovaný za podmínek definovaných v OBE_6.0.	Závazný	1
CIS_9.0	Změna varianty ve verzi	Systém umožňuje instanci objektu Číselník změnit variantu za verzi za podmínek definovaných v obecném požadavku (viz OBE_7.0).	Závazný	1
CIS_11.0	Dynamické atributy objektu Číselník	Systém umožňuje uživateli definovat další atributy Číselníku podle byznys pravidel definovaných v kapitole 3.6.2 Dynamické atributy objektu Položka číselníku.	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Takto přidané atributy jsou zároveň atributem každé Položky číselníku.		
CIS_12.0	Atributy pro správu objektu a pro schvalování lokálního číselníku.	Systém umožňuje uživateli nastavit všechny atributy na požadované hodnoty a přiřadit danému Číselníku požadované uživatelské místo. Podrobněji viz kapitola 3.5.1 Atributy objektu Číselník.	Závazný	2
CIS_13.0	Vytvoření Položky číselníku	Systém umožňuje uživateli vytvořit novou Položku číselníku objektu Číselník podle pravidel definovaných v kapitole 3.6 Objekt Položka číselníku.	Závazný	1
CIS_13.1	Jednoznačnost kódu objektu Položka číselníku	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Položka číselníku je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Položka číselníku, které se vztahují ke shodné instanci objektu Číselník.	Závazný	1
CIS_14.0	Vytvoření Položky číselníku ručním typování	Systém umožňuje uživateli přidávat položky do objektu Číselník ve stavu Projektovaný ručním typováním.	Závazný	1
CIS_15.0	Vytvoření strukturovaného souboru pro import Položek číselníků	Systém umožňuje vkládání nebo aktualizaci Položek číselníku prostřednictvím externího souboru. Systém umožňuje uživateli vyexportovat do externího souboru strukturu konkrétního Číselníku. Strukturu souboru (zjednodušeně řečeno jeho sloupce) tvoří jednotlivé atributy daného Číselníku. Dále soubor obsahuje sloupec pro zachycení hierarchické struktury mezi jednotlivými Položkami číselníku. Takto vytvořený soubor je pak uživatelem naplněn datovým obsahem mimo systém SDAT. Detailní specifikace struktury požadovaného externího souboru je	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		předmětem řešení SDAT (tj. nejedná se o existující strukturu, kterému by bylo řešení SDAT nutno přizpůsobit). Požadované formáty: a) XLS – atributy jako hlavičky sloupců v pořadí odpovídající datovému modelu objektu Položka číselníku, b) XML – atributy jako XML elementy.		
CIS_16.0	Import Položek číselníku z externího souboru	Systém umožňuje uživateli vložit nebo aktualizovat Položky číselníku prostřednictvím souboru vytvořeného podle CIS_15.0. Atributem, přes který probíhá porovnání obsahu souboru a obsahu objektu Položka číselníku, je kód položky. Import má dvě fáze: 1) diagnostika: systém provede diagnostiku importu. Uživatel má možnost se rozhodnout, zda import provést nebo ho zrušit a vrátit se k úpravě souboru. Systém oznámí uživateli: i. položky, které budou přidány, ii. položky, které budou aktualizovány, iii. položky, kterým bude ukončena platnost,, iv. zda číselník obsahuje hierarchii položek. 2) samotný import: uživatel má možnost se rozhodnout, jaké operace (i. – iv.) importem souboru provede: i. přidání nových Položek číselníku –ano/ne, ii. aktualizaci hodnot atributů (kromě klíčového atributu kód položky) – ano/ne, iii. ukončení platnosti položek – ne, iv. vytvoření Hierarchie číselníku – ano/ne. V případě vytváření Hierarchie číselníku systém vytvoří instanci	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Hierarchie číselníku s generickým názvem a následně vytvoří Položky hierarchie na základě vztahů mezi Položkami číselníku v importovaném souboru. Bod iii lze provést za podmínek definovaných v požadavku CIS_19.0. Bod iv. lze provést, pouze pokud byly provedeny všechny předchozí body a při zachování podmínek dle kapitoly 3.7.2 Proces tvorby Hierarchií číselníku. Při tomto způsobu vytváření Hierarchie číselníku musí být zároveň dodržena všechna pravidla podle kapitoly 7.11 Hierarchie číselníku, Položka hierarchie.		
CIS_17.0	Změna Položky číselníku	Systém umožňuje změnit atributy u existující Položky číselníku podle pravidel definovaných v kapitole 2.3 Vazby mezi jednotlivými objekty	Závazný	1
CIS_18.0	Smazání Položky číselníku	Systém umožňuje smazat Položku číselníku podle pravidel uvedených v kapitole 2.4.5.4 Smazání objektů. Smazání Položky číselníku je povoleno jen za následujících podmínek: • Číselník musí být ve stavu Projektovaný, • Položka číselníku nesmí být použita v Konkretizovaném parametru, • Položka číselníku nesmí být použita v Převodníku, • Položka číselníku nesmí být použita v žádné Doméně číselníku, • nesmí existovat vazba Položky číselníku na Položku hierarchie.	Závazný	1
CIS_19.0	Ukončení platnosti Položky číselníku	Systém umožňuje uživateli ukončit Položku číselníku podle pravidel uvedených v kapitole 2.4.5.3 Ukončování platnosti objektů. Ukončení platnosti Položky číselníku je povoleno za stejných podmínek jako v CIS_12.0.	Závazný	1
CIS_20.0	Označení součtové Položky	Systém označí součtovou Položku číselníku v seznamu Položek číselníku znakem ∑ v závislosti na tom, zda existuje vazba na uzlovou položku	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	číselníku v Číselníku	objektu Hierarchie číselníku.		
CIS_21.0	Zobrazení elementárního rozkladu součtové Položky číselníku v Číselníku	Systém umožňuje uživateli zobrazit elementární Položky číselníku pro součtovou Položku číselníku přímo ze seznamu Položek číselníku.	Závazný	1
CIS_22.0	Základní zobrazení Položek číselníku	Systém umožňuje uživateli zobrazit pořadí Položek číselníku v základním abecedně řazeném seznamu podle atributu kód položky.	Závazný	1
CIS_23.0	Uživatelské řazení Položek číselníku	Systém umožňuje uživateli definovat ve formuláři pořadí Položek číselníku podle uživatelského řazení pořadí atributů. Např. pro Číselník měn je definován dynamický atribut „země-měna“ a uživatel bude chtít seřadit položky podle zemí, kde je platná stejná měna. Jako první atribut pořadí si zvolí dynamický atribut „země-měna“ (abecedně) a Položky číselníku se přeřadí podle zadání uživatele.	Závazný	2
CIS_24.0	Export Položek číselníku	Systém umožňuje uživateli vyexportovat Položky číselníku, zobrazené dle CIS_22.0 nebo CIS_23.0, do zvoleného formátu (xls, pdf, xml).	Závazný	2
CIS_25.0	Tisk Číselníku	Systém umožňuje uživateli tisk zobrazeného Číselníku, a to v pořadí podle požadavku C_22.0 nebo C_23.0. Požadavek na tisk zobrazeného Číselníku lze řešit ve dvou krocích (export do formátu umožňující tisk a následně vlastní tisk). Tento princip lze použít obecně i v případě dalších obdobných požadavků na tisk ze systému SDAT.	Závazný	2
CIS_26.0	Volitelné	Systém umožňuje uživateli nastavit parametricky, které atributy mají být	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	parametry tisku Číselníku s položkami	vytištěny.		
CIS_27.0	Hlídání duplicit	Systém neumožňuje vytvoření duplicitní Položky číselníku v rámci jednoho Číselníku (duplicitní kód i název).	Závazný	1

7.11 Hierarchie číselníku, Položka hierarchie

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
HIE_1.0	Vytvoření Hierarchie číselníku	Systém umožňuje vytvořit novou prázdnou instanci objektu Hierarchie číselníku za podmínek definovaných v kapitole 3.7 Objekt Hierarchie číselníku. Při vytvoření nové Hierarchie číselníku systém automaticky vytvoří novou verzi instance objektu Hierarchie číselníku a nastaví jí stav na Projektovaný. Podrobně je popis životního cyklu instancí popsán v kapitole 2.2.6 Přístup „Sledování historie – časová platnost + stavy“. Systém kontroluje název a kód vytvářené Hierarchie číselníku. Pokud indikuje shodu s již existující Hierarchií číselníku, nepovolí uživateli vytvořit novou instanci Hierarchie číselníku.	Závazný	1
HIE_1.1	Jednoznačnost kódu objektu Hierarchie číselníku	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Hierarchie číselníku je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Hierarchie číselníku, které se vztahují ke shodné instanci objektu Číselník.	Závazný	1
HIE_2.0	Editace atributů	Systém umožňuje měnit hodnoty atributů existující instance Hierarchie	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	Hierarchie číselníku	číselníku za splnění byznys podmínek uvedených v kapitole 3.7 Objekt Hierarchie číselníku. Systém umožňuje uživateli rozhodnout se pro založení verze nebo varianty pro případ editace hodnot atributů objektu. Při provádění změn v hodnotách atributů systém kontroluje, zda změny jsou v souladu s nastavením systému. Pravidla pro změnu atributů jsou podrobně popsána v kapitole 2.3 Vazby mezi jednotlivými objekty. Pokud není změna atributu v nějakém stavu možná, pak systém uživateli tuto změnu vůbec nedovolí. Tento funkční požadavek se nevztahuje k editaci „atributu pro správu objektu“ a „atributu pro provedení synchronizace“.		
HIE_3.0	Nastavení atributu pro správu Hierarchie číselníku	Nastavení „atributu pro správu objektu“ je zděděno od zdrojového Číselníku.	Závazný	2
HIE_4.0	Nastavení UM pro správu objektu Hierarchie číselníku	Nastavení uživatelského místa pro správu (vytvoření objektu, plnění objektu Položkami hierarchie, smazání objektu, ukončení platnosti objektu) je zděděno od nastavení „atributu pro správu objektu“ zdrojového Číselníku.	Závazný	2
HIE_5.0	Smazání instance objektu Hierarchie číselníku	Systém umožňuje smazat instanci objektu Hierarchie číselníku (viz OBE_1.0).	Závazný	1
HIE_6.0	Plnění objektu Hierarchie	Systém umožňuje uživateli plnit nebo změnit obsah instance objektu Hierarchie číselníku za následujících podmínek:	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	číselníku Položkami hierarchie	• existuje vytvořená verze instance objektu Hierarchie číselníku ve stavu Projektovaný (ve variantě instance objektu Hierarchie číselníku systém nedovolí uživateli plnit ani modifikovat obsah existující Hierarchie číselníku), • všechny ostatní Hierarchie číselníku, které vycházejí ze stejného Číselníku jako vytvářená/měněná Hierarchie číselníku, mají nastaven „atribut provedení synchronizace“ na hodnotu „ano“ Podrobně je proces tvorby instance objektu Hierarchie číselníku popsán v kapitole 3.7.2 Proces tvorby Hierarchií číselníku.		
HIE_7.0	Aktualizace Hierarchie číselníku	Proces aktualizace instance objektu Hierarchie je popsán podrobně v kapitole 3.7.2.1 Aktualizace instance Hierarchie číselníku. Systém povolí uživateli provést aktualizaci, pokud má objekt založenu verzi objektu ve stavu Projektovaný. Pokud není úspěšně dokončen proces aktualizace (např. nastane nesoulad mezi Doménami číselníku vzniklými z Hierarchie číselníku a aktualizovanou Hierarchií číselníku), pak dochází k nekonzistenci a uživatel je vyzván k vyřešení této nekonzistence.	Závazný	1
HIE_8.0	Synchronizace Hierarchie číselníku	Proces synchronizace je možné spustit až po úspěšném dokončení procesu aktualizace. Proces synchronizace instance objektu Hierarchie číselníku je popsán podrobně v kapitole 3.7.2.2 Synchronizace instance Hierarchie číselníku.	Závazný	1
HIE_9.0	Pravidla pro řízení aktualizace a synchronizace Hierarchie	Proces aktualizace a synchronizace je řízen pravidly definovanými v kapitole 8.2 Příloha 2 - Příklady pro pravidla aktualizace a synchronizace Hierarchií číselníku vč. vzorových příkladů. Chování systému podle pravidel při aktualizaci a synchronizaci je řízeno	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	číselníku	algoritmy zabudovanými do systému.		
HIE_10.0	Diagnostika dopadů při aktualizaci Hierarchie číselníku	Systém zobrazuje uživateli diagnostiku dopadu změn na všechny související objekty v důsledku změny prováděné v objektu Hierarchie číselníku. V rámci diagnostiky jsou uživateli zobrazovány následující informace: • dopad na přímo související objekty, tj. Domény číselníku, které jsou vytvořeny přímo z aktualizované Hierarchie, • dopad na nepřímo související objekty, tj. objekty které používají přímo související objekty (např. Datová oblast, kde je použita Doména číselníku vytvořená z aktualizované Hierarchie číselníku),	Závazný	1
HIE_10.1	Diagnostika dopadů při synchronizaci Hierarchie číselníku	Systém zobrazuje uživateli diagnostiku dopadu změn na všechny související objekty v důsledku změny prováděné v objektu Hierarchie číselníku. V rámci diagnostiky jsou uživateli zobrazovány následující informace: • dopad na přímo související objekty, tj. na ostatní Hierarchie číselníku v souvisejících uzlech s vyznačením dopadu do měněného objektu, • dopad na Domény číselníku, které jsou vytvořeny z Hierarchií číselníku, které budou změnou v aktualizované Hierarchii číselníku dotčeny, • dopad na nepřímo související objekty, které obsahují měněné přímo související objekty s vyznačením dopadu do měněných objektů a dopadu na změnu stavu dotčeného objektu.	Závazný	1
HIE_11.0	Export diagnostiky dopadu do souboru	Systém umožňuje uživateli provést export diagnostiky dopadu, jak u aktualizace, tak i u synchronizace, do textového souboru (pdf).	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
HIE_12.0	Diagnostika konfliktů při synchronizaci Hierarchie číselníku – systémem neřešitelný konflikt	Pokud systém při procesu synchronizace zjistí, že nastala situace, kdy jsou porušena pravidla tvorby Hierarchie číselníku, oznámí uživateli konflikt a v rámci diagnostiky zobrazí uživateli očekáváný dopad na přímo související objekty a vyzve uživatele k vyřešení konfliktu. Situace, kdy je vyhodnocen systémem neřešitelný konflikt, jsou popsány v kapitole 8.2.9 Konflikt typu Duplicita v nezávislých uzlech resp. 8.2.10 Konflikt typu Duplicita v závislých uzlech. V případě obou výše uvedených situací není synchronizace dokončena a uživateli je zobrazena informace o tom, že situace nelze vyřešit.	Závazný	1
HIE_13.0	Diagnostika konfliktů při synchronizaci Hierarchie číselníku – systémem řešitelný konflikt	Pokud systém při procesu synchronizace zjistí, že nastala situace, že položka má změnit svojí nadřízenou položku, systém uživateli umožňuje potvrdit změnu, pokud je požadována, a synchronizace je dokončena.	Závazný	1
HIE_14.0	Propagace změn do souvisejících objektů při aktualizaci	Systém propaguje změny způsobené aktualizací instance objektu Hierarchie číselníku do přímo souvisejících objektů následovně: • do objektů Knihovny, do Metodiky vykazovacího rámce za podmínky, že platnost dotčených objektů leží v intervalu platnost_od Hierarchie číselníku až platnost_do Hierarchie číselníku, • pro objekty Knihovny, když platnost_od Hierarchie číselníku je rovna platnost_od verze Domény číselníku: • jestliže je v rámci aktualizace dotčena změnou Doména číselníku, která je ve stavu Projektovaný, systém provede její změnu okamžitě,	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		• jestliže je v rámci aktualizace dotčena Doména číselníku ve stavu Platný, systém vytvoří Doméně číselníku novou verzi a provede změnu v nové verzi. Pokud je doména použita ve verzi Výkazu, resp. ve verzi Datové oblasti ve stavu Platný, bude dotčený Výkaz a Datová oblast zaverzovány v následné Metodice vykazovacího rámce. Pokud toto verzování není v následné Metodice provedeno, pak kontrola konzistence ohlásí nesoulad. • jestliže je v rámci aktualizace dotčena Doména ve stavu Schválený, systém vrátí doménu do stavu Projektovaný a provede změnu. • pro objekty Knihovny, když platnost_od Hierarchie číselníku je větší než platnost_od verze Domény číselníku je Doméně číselníku vždy vytvořena nová verze s platnost_od Domény číselníku shodné s platnost_od Hierarchie číselníku a předchozí verzi Domény číselníku je zkrácena platnost_od na datum platnost_od -1. Pro objekty v Metodice vykazovacího rámce platí: • do objektů, které jsou ve stavu Projektovaný, jsou změny vyplývající ze změn v aktualizované Hierarchii číselníku promítány okamžitě a uživatel objektu je o tomto informován, • do objektů, které jsou ve stavu Schválený, nejsou změny promítány okamžitě, uživateli je zaslána informace a uživatel má povinnost avizované změny provést, jinak nastává stav nekonzistence, • do objektů, které jsou ve stavu Platný jsou změny promítány tak, že je nadřízeným objektům vytvořena nová verze a změna je promítnuta v následné Metodice vykazovacího rámce (není-li již vytvořena).		
HIE_15.0	Propagace změn do souvisejících	Pro objekty Knihovny platí totéž jako v HIE_14.0 pro Domény číselníku. Pro související Hierarchie číselníku platí následující:	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	objektů při synchronizaci	• do souvisejících Hierarchií číselníku ve stavu Projektovaný jsou změny promítány okamžitě, • související Hierarchie číselníku ve stavu schváleném jsou vráceny do stavu Projektovaný a změny jsou promítnuty, souvisejícím Hierarchiím číselníku ve stavu Platný je vytvořena nová verze se stavem Projektovaný a nastaveno platnost_od jako platnost_od Hierarchie číselníku, kterou je změna vyvolána. Změny jsou do nové verze instance objektu pak promítnuty. Pro objekty v Metodice vykazovacího rámce platí totéž jako v HIE_14.0.		
HIE_16.0	Prostředí pro projektování Hierarchie číselníku	Systém umožňuje grafické prostředí pro projektování objektu Hierarchie číselníku, v němž má uživatel možnost: • opticky rozlišit jednotlivé úrovně Hierarchie číselníku, • označovat souvislé části Hierarchie číselníku, • vytvářet Hierarchie číselníku z Položek hierarchie ostatních Hierarchií číselníku, • přesouvat celé uzle z jedné úrovně do druhé, • vkládat Položky číselníku pod označenou Položku hierarchie na stejnou úroveň nebo o úroveň níže, • mazat celé uzle Hierarchie číselníku.	Závazný	1
HIE_16.1	Podpůrné funkce pro projektování Hierarchie číselníku	Systém umožňuje v rámci uživatelského prostředí používat následující funkce: • vyhledávání Položek hierarchie v určité Hierarchii číselníku, • možnost zobrazení jen určité úrovně, • tisk Hierarchie číselníku s grafickým vyznačením úrovní,	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		• zobrazení souvisejících uzlů v ostatních Hierarchiích číselníku, • zobrazení diagnostiky dopadu při aktualizaci a synchronizaci Hierarchie číselníku, • export diagnostiky dopadu do textového souboru.		
HIE_17.0	Smazání instance Hierarchie číselníku včetně jejího obsahu tj. instancí Položek hierarchie	Systém umožňuje smazání instance objektu Hierarchie číselníku (viz OBE_1.0), pokud je splněna podmínka, že uzlová položka Hierarchie (která je jedinečná – tzn. je jen v Hierarchii číselníku, které je právě mazána) není použita pro konkretizaci Ukazatele.	Závazný	1
HIE_18.0	Ukončení platnosti Hierarchie číselníku	Systém umožňuje ukončení platnosti instance objektu Hierarchie (viz OBE_ 2.0 a OBE_2.1).	Závazný	1
HIE_19.0	Zobrazení Hierarchie číselníku	Systém umožňuje uživateli zobrazit Hierarchii číselníku tak, aby byly graficky rozlišeny jednotlivé úrovně.	Závazný	1
HIE_20.0	Export Hierarchie číselníku	Systém umožňuje uživateli exportovat celou Hierarchii číselníku do souboru (pdf, xls, xml).	Závazný	2
HIE_21.0	Tisk Hierarchie číselníku	Systém umožňuje vytištění celé Hierarchie číselníku a jejích atributů včetně informace o verzi zdrojového Číselníku, ze kterého je Hierarchie číselníku vytvořena.	Závazný	2
HIE_23.0	Atribut pro provedení synchronizace	Systém umožňuje nastavení atributu pro provedení synchronizace v závislosti na dokončení akce synchronizace. Pokud je synchronizace Hierarchie číselníku dokončena, systém u této Hierarchie číselníku uživateli zobrazí, že synchronizaci provedl.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		V seznamu Hierarchií číselníku jsou označeny všechny synchronizované Hierarchie číselníku. Nesynchronizovaná Hierarchie číselníku je opticky rozlišena. Pokud existuje nějaká nesynchronizovaná Hierarchie číselníku v rámci Číselníku, pak systém nepovolí aktualizaci další Hierarchie číselníku. V takovém případě oznámí uživateli kód nesynchronizované Hierarchie číselníku.		
HIE_24.0	Hlídání duplicit	Systém neumožňuje uživateli vytvořit instanci objektu Hierarchie číselníku se stejným kódem a názvem v rámci jednoho Číselníku.	Závazný	1
HIE_25.0	Ukončení platnosti instance objektu Hierarchie číselníku systémem v závislosti na zplatnění jiné verze/varianty téhož objektu	Systém automaticky ukončuje platnost instanci objektu Hierarchie číselníku (viz OBE_2.1).	Závazný	1
HIE_26.0	Prodloužení platnosti instance objektu Hierarchie číselníku	Systém umožňuje uživateli prodloužit platnost instance objektu Hierarchie číselníku (viz OBE_3.0).	Závazný	1
HIE_27.0	Schválení instance objektu	Systém umožňuje uživateli schválit instanci objektu Hierarchie číselníku (viz OBE_4.0).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	Hierarchie číselníku			
HIE_28.0	Zplatnění instance objektu Hierarchie číselníku	Systém umožňuje uživateli zplatnit instanci objektu Hierarchie číselníku (viz OBE_ 5.0).	Závazný	1
HIE_29.0	Změna stavu instance objektu Hierarchie číselníku ze stavu Schválený na stav Projektovaný	Systém umožňuje uživateli změnit stav instance objektu Hierarchie číselníku ze stavu Schválený na stav Projektovaný (viz OBE_6.0).	Závazný	1
HIE_30.0	Změna varianty na verzi u Hierarchie číselníku	Systém umožňuje uživateli změnit vytvořenou variantu instance objektu Hierarchie číselníku na verzi instance objektu Hierarchie číselníku (viz OBE_7.0).	Závazný	1

7.12 Doména číselníku

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DOC_1.0	Vytvoření prázdné instance ruční Doména číselníku	Systém umožňuje vytvoření instance objektu Doména číselníku podle byznys pravidel definovaných v kapitole 3.9 Objekt Doména číselníku. Při vytvoření nové instance je automaticky založena nová verze Domény číselníku. Podrobně je popis životního cyklu popsán v 2.2.6 Přístup „Sledování historie – časová platnost + stavy“.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Vytvoření prázdné instance je prováděno uživatelem v případě ruční Domény číselníku.		
DOC_2.0	Vytvoření automatické Domény číselníku	Systém umožňuje vytvoření automatické Domény číselníku z Hierarchie číselníku a to následovně: • uživatel vybere konkrétní uzel Hierarchie číselníku a spustí funkci pro vytvoření automatické Domény číselníku, jejíž položky jsou tvořeny n-1 úrovní dané Hierarchie číselníku (následující nižší úrovní Hierarchie číselníku). Takto vytvořená Doména číselníku je jen jednoúrovňová, • systém takto vytvořené doméně přiřadí automaticky standardní atributy a vyplní je následovně: o název objektu - je stejný jako název uzlové položky, o kód objektu – je odvozen od kódu uzlové položky a to následovně A_+“kód uzlové položky“_pořadové číslo, o datum_vytvoření - datum, kdy uživatel Doménu číselníku vytvořil, o platnost_od a platnost_do - je odvozena od zdrojové instance objektu Hierarchie číselníku, • ostatní atributy nejsou systémem plněny. Možnost jejich vyplnění je na uživateli, • ostatní základní atributy jsou nastavovány dle popisu uvedených v kapitole 2.4.2 Standardní atributy objektů.	Závazný	1
DOC_2.1	Jednoznačnost kódu objektu Doména číselníku	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Doména číselníku je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Doména číselníku, které se vztahují ke shodné instanci objektu Číselník.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DOC_3.0	Editace hodnot atributů ruční Domény číselníku	Systém umožňuje měnit hodnoty atributů existují instance objektu ruční Doména číselníku za splnění byznys podmínek uvedených v kapitole 2.3 Vazby mezi jednotlivými objekty. Uživatel má možnost se rozhodnout, zda vytvoří novou verzi nebo variantu instance objektu. Systém hlídá, zda provedené změny jsou v souladu s nastavením systému. Podrobně jsou pravidla uvedena v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
DOC_4.0	Editace atributů automatické Domény číselníku	Systém umožňuje uživateli definovat hodnoty atributů popis objektu a poznámka. Hodnoty atributů vyplněné systémem není povoleno editovat uživatelem.	Závazný	1
DOC_5.0	Nastavení atributu Typ Domény číselníku	Systém eviduje podle způsobu vytvoření instance objektu Doména číselníku její typ, tj. **automatická** nebo **ruční** a podle evidovaného typu povolí úpravy hodnot atributů podle funkčních požadavků DOC_4.0 nebo DOC_3.0. Nastavení hodnoty atributu viz kapitola 3.9 Objekt Doména číselníku.	Závazný	1
DOC_6.0	Nastavení atributu Způsob aktualizace domény	Systém umožňuje nastavit atribut „způsob aktualizace domény“ na základě způsobu vytvoření instance objektu Doména číselníku. Popis plnění atributu je podrobně popsán v kapitole 3.9.1 Atributy objektu Doména číselníku.	Závazný	1
DOC_7.0	Hlídání použití domény při projektování	Systém podle typu Domény číselníku hlídá použití Domény číselníku pro projektování výkazů, a to následovně: a) ruční Doména číselníku: • lze použít pro projektování statických a kartotékových Datových oblastí, • pro dynamické Datové oblasti je použití ruční Domény číselníku	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		omezeno jen na deklaraci oboru hodnot instance objektu Ukazatele a rozgenerování položek Domény číselníku v ose, která není deklarována jako dynamická, b) automatická Doména číselníku - lze použít pro projektování dynamických Datových Oblastí pro dynamicky deklarované Parametry nad Číselníkem.		
DOC_8.0	Duplicitní Domény číselníku	Systém podle elementárního obsahu indikuje duplicitní Domény číselníku a informuje uživatele tak, že mu zobrazí kód a název duplicitních Domén číselníku. Zároveň uživateli zobrazí i obsah Domény.	Závazný	1
DOC_9.0	Zobrazení obsahu Domény číselníku	Systém umožňuje uživateli zobrazit: a) seznam elementárních Položek číselníku v plochém rozlišení, b) seznam součtových i elementárních Položek číselníku v plochém nebo hierarchickém pořadí podle způsobu vytvoření, c) pokud se jedná o Doménu číselníku odvozenou z Hierarchie číselníku, je zobrazen i odkaz na zdrojovou Hierarchii číselníku.	Závazný	1
DOC_10.0	Způsob aktualizace Domény číselníku vytvořené z Hierarchie číselníku	Systém modifikuje při aktualizaci instance objektu Doména číselníku vytvořené z Hierarchie číselníku Domény číselníku podle pravidel popsaných v číslovaných položkách 1 až10 v kapitole 3.9 Objekt Doména číselníku a příloze 8.3 Příloha 3 - Příklady pro pravidla aktualizace Domén číselníku.	Závazný	1
DOC_11.0	Smazání ruční Domény číselníku	Systém umožňuje uživateli smazat Doménu číselníku (viz OBE_1.0).	Závazný	1
DOC_12.0	Smazání automatické Domény číselníku	Systém umožňuje smazání uživatelem vytvořené automatické Domény číselníku. Toto smazání nemá žádný dopad na Hierarchii číselníku, ze které automatická Doména číselníku vznikla.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Smazání je umožněno za podmínky, že automatická Doména číselníku není použita pro konkretizaci Parametru v dynamické Datové oblasti, který tvoří dimenzi.		
DOC_13.0	Ukončení platnosti ruční Domény číselníku	Systém umožňuje ukončení platnosti ruční domény podle byznys pravidel uvedených v OBE_2.0 a OBE_2.1.	Závazný	1
DOC_14.0	Ukončení platnosti automatické Domény	Systém umožňuje ukončení platnosti automatické domény číselníku (viz OBE_2.0 a OBE_2.1).	Závazný	1
DOC_17.0	Tisk Domény číselníku	Systém umožňuje uživateli vytisknout jednotlivou Doménu číselníku, skupinu Domén číselníku nebo všechny Domény číselníku podle parametrů nastavených na začátku tisku. Parametry tisku jsou: • Číselník, více Číselníků, všechny Číselníky, • interval časové platnosti, • jednotlivé Domény číselníku, více Domén číselníku, všechny Domény číselníku. Pro parametrizaci tiskových sestav systém umožňuje použít tzv. „wild cards“. Na tiskové sestavě je uveden zdrojový Číselník, verze Domény číselníku a interval její platnosti.	Závazný	2
DOC_18.0	Export Domény číselníku	Systém umožňuje export Domény číselníku do textového souboru ve zvoleném parametrickém rozsahu jako DOC_17.	Závazný	2
DOC_19.0	Prodloužení platnosti instance	Systém umožňuje uživateli prodloužit platnost instance objektu Doména číselníku (viz OBE_3.0).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	objektu Doména číselníku			
DOC_20.0	Schválení instance objektu Doména číselníku	Systém umožňuje uživateli převést instanci objektu Doména číselníku ze stavu Projektovaný do stavu Schválený (viz OBE_4.0).	Závazný	1
DOC_20.1	Hromadné schválení Domén číselníku	Systém podporuje hromadné schválení Domén číselníku v rámci jednoho Číselníku.	Závazný	1
DOC_21.0	Zplatnění instance objektu Doména číselníku	Systém umožňuje zplatnit instanci objektu Číselník (viz OBE_5.0).	Závazný	1
DOC_21.1	Hromadné zplatnění Domén číselníku	Systém podporuje hromadné zplatnění Domén číselníku v rámci jednoho Číselníku.	Závazný	1
DOC_22.0	Změna stavu instance objektu Doména číselníku ze stavu schválený na stav Projektovaný (u instancí	Systém umožňuje uživateli změnit stav instance objektu Doména číselníku ze stavu Schválený na stav Projektovaný (viz OBE_6.0).	Závazný	1
DOC_23.0	Změna varianty na verzi u Domény číselníku	Systém umožňuje uživateli změnit vytvořenou variantu instance objektu Doména číselníku na verzi instance objektu Doména číselníku (viz OBE_7.0).	Závazný	1

7.13 Převodník, Položka převodníku

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
PŘE_1.0	Zobrazení Převodníků a Převodníkových položek	Systém umožňuje zobrazení náhledu na instance objektů Převodník a Položka převodníku tak, že zobrazuje seznam všech definovaných Převodníků (instancí objektu Převodník) a pro uživatelem vybranou instanci objektu Převodník zobrazí seznam všech převodníkových položek (související instance objektu Položka převodníku). Systém v seznamu Převodníků automaticky zobrazuje nejaktuálnější verzi a variantu instance objektu Převodník, nicméně umožňuje získat informace i o všech jeho předcházejících verzích/variantách. Ze zobrazení jsou patrné zejména následující informace: • zobrazení kompletní historie všech verzí a variant, včetně vymezení časové oblasti každé verze/varianty, kterými daný Převodník prošel během svého životního cyklu, • zobrazení kompletní sady stavů, kterými prošla ta která verze/varianta Převodníku, • pro každý Převodník je zobrazeno, z jakých číselníků je tvořen (vazby zdrojový číselník/cílový číselník v objektovém modelu), • pro každou Položku převodníku je zobrazeno, z jakých číselníkových položek je tvořena (vazba zdrojová položka/cílová položka v objektovém modelu).	Závazný	3
PŘE_2.0	Vytvoření Převodníku	Systém umožňuje vytvořit nový Převodník (instanci objektu Převodník) za splnění byznys podmínek uvedených v kapitole 3.10 Objekt Převodník. Při vytvoření nového Převodníku systém automaticky založí novou verzi Převodníku a automaticky tuto verzi založí ve stavu Projektovaný. Podrobně je popis životního cyklu instancí popsán v kapitole 2.2 Sledování historie instancí objektů.	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
PŘE_2.1	Jednoznačnost kódu objektu Převodník	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Převodník je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Převodník. Systém nepovoluje změnit kód objektu na hodnotu již použitou u jiné instance objektu Převodník. Kód objektu lze změnit jen v případě, že existuje pouze první verze, a to ve stavu Projektovaný, a není nikde použitá.	Závazný	3
PŘE_3.0	Editace Převodníku	Systém umožňuje měnit atributy existujícího Převodníku (existující instance objektu Převodník) za splnění byznys podmínek uvedených v kapitole 3.10 Objekt Převodník. Před zahájením editace vybraného Převodníku systém umožňuje uživateli se rozhodnout, zda má být založena nová verze, varianta nebo zda má být provedena změna v rámci existující verze/varianty. Po provedení změn systém kontroluje, zda jsou tyto změny v souladu s nastavením systému, který definuje, jak mohou být jednotlivé atributy měněny. Podrobně jsou pravidla pro měnění jednotlivých atributů popsány v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	3
PŘE_4.0	Smazání Převodníku	Systém umožňuje smazat jakýkoli existující převodník bez ohledu na jakékoli další okolnosti v souladu s funkčním požadavkem OBE_1.0. Pokud dojde ke smazání instance objektu Převodník, dojde zároveň ke smazání všech existujících instancí objektu Položka převodníku. Smazání instance objektu Položka převodníku nemá žádný vliv na instance objektu Položka číselníku.	Závazný	3
PŘE_5.0	Ukončení časové platnosti Převodníku	Systém umožňuje ukončit časovou platnost vybrané verze Převodníku v souladu s funkčním požadavkem OBE_2.0. Podrobně je postup ukončování platnosti verze popsán v 2.2.6 Přístup „Sledování historie – časová platnost + stavy“.	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Zároveň musí dojít k ukončení časové platnosti všech podřízených souvisejících instancí objektu Položka převodníku tak, aby jejich časová platnost nevybočila z rozsahu nadřízené instance objektu Převodník.		
PŘE_6.0	Vytvoření Položky převodníku	Systém umožňuje vytvořit novou Položku převodníku v rámci vybraného převodníku za splnění byznys podmínek definovaných v kapitole 3.11 Objekt Položka převodníku. Vytvořit novou Položku převodníku je možno bez ohledu na to, v jakém stavu se nachází nadřízená instance objektu Převodník. Je však nutné, aby systém zohlednil nastavení závislosti mezi objekty (viz kapitola 2.3 Vazby mezi jednotlivými objekty), a podle toho buď vytvořil novou verzi nebo variantu nebo neprovedl žádný zásah do existující instance objektu Převodník. Položky převodníku lze vytvořit i hromadně.	Závazný	3
PŘE_7.0	Změna Položky převodníku	Systém umožňuje změnit existující Položku převodníku. V rámci této akce je možné změnit pouze zdrojovou a cílovou Položku převodníku. Akce může být dokončena jen tehdy, pokud jsou splněny všechny byznys podmínky definované v kapitole 2.3 Vazby mezi jednotlivými objekty. Změnit existující položku převodníku je možno bez ohledu na to, v jakém stavu se nachází nadřízená instance objektu Převodník, je však nutné, aby systém zohlednil nastavení závislosti mezi objekty (viz kapitola 2.3 Vazby mezi jednotlivými objekty), a podle toho buď vytvořil novou verzi nebo variantu nebo neprovedl žádný zásah do existující instance objektu Převodník. Součástí změny Položky převodníku může být změna časové platnosti (nastavení hodnot atributů platnost_od a platnost_do). Při změně této časové platnosti musí být zajištěno splnění pravidla, že časová platnost	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		instance objektu Položka převodníku nesmí vybočovat z časové platnosti nadřízené související instance objektu Převodník		
PŘE_8.0	Smazání Položky převodníku	Systém umožňuje smazat existující Položku převodníku bez ohledu na jakékoli další okolnosti. Smazat existující položku převodníku je možno bez ohledu na to, v jakém stavu se nachází nadřízená instance objektu Převodník, je však nutné, aby systém zohlednil nastavení závislosti mezi objekty (viz kapitola 2.3 Vazby mezi jednotlivými objekty), a podle toho buď vytvořil novou verzi nebo variantu nebo neprovedl žádný zásah do existující instance objektu Převodník.	Závazný	3
PŘE_9.0	Změna stavu Převodníku ze Schválený na Projektovaný	Systém umožňuje provést návrat ke stavu Projektovaný ze stavu Schválený v souladu s pravidly definovanými v OBE_6.0 u každé instance objektu Převodník.	Závazný	3
PŘE_10.0	Zplatnění nové verze/varianty Převodníku	Systém umožňuje zplatnění nové verze Převodníku v souladu s pravidly definovanými v požadavku OBE_5.0.	Závazný	3

7.14 Účtová osnova, účet

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
UCO_1.0	Zobrazení Účtové osnovy a účtů	Systém umožňuje zobrazení náhledu na instance objektů Účtová osnova a Účet tak, že zobrazuje seznam všech definovaných účtových osnov a pro uživatelem vybranou instanci objektu Účtová osnova zobrazí seznam všech účtů v hierarchii.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
UCO_2.0	Vytvoření účtové osnovy	Systém umožňuje uživateli vytvořit novou Účtovou osnovu (první instanci objektu Účtová osnova) za splnění byznys podmínek uvedených v kapitole 3.12 Objekt Účtová osnova. Při vytvoření nové Účtové osnovy systém automaticky vytvoří novou verzi a automaticky tuto verzi založí ve stavu Projektovaný.	Závazný	1
UCO_2.1	Jednoznačnost kódu objektu Účtová osnova	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Účtová osnova je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Účtová osnova. Systém nepovoluje změnit kód objektu na hodnotu již použitou u jiné instance objektu Účtová osnova. Kód objektu lze změnit jen v případě, že existuje pouze první verze, a to ve stavu Projektovaný, a není nikde použitá.	Závazný	1
UCO_3.0	Editace Účtové osnovy	Systém umožňuje měnit atributy existující Účtové osnovy (existující instance) za splnění byznys podmínek uvedených v kapitole 3.12 Objekt Účtová osnova. Před zahájením editace systém umožňuje uživateli se rozhodnout, zda má být založena nová verze/varianta nebo změna bude provedena v rámci existující verze/varianty. Systém kontroluje, zda prováděné změny jsou v souladu s nastavením systému, které definuje, jak mohou být jednotlivé atributy měněny.	Závazný	1
UCO_3.1	Změna varianty Účtové osnovy ve verzi	Systém umožňuje uživateli změnit variantu Účtové osnovy ve verzi (viz OBE_7.0).	Závazný	1
UCO_4.0	Smazání Účtové osnovy	Systém umožňuje uživateli smazat Účtovou osnovu ve stavu Projektovaný (viz OBE_1.0). Pokud v mazané instanci vznikly nové Účty nebo proběhla editace Účtů, tyto změny jsou smazány spolu s Účtovou	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		osnovou. Pokud je mazaný Účet použit, dojde ke zrušení vazby tohoto Účtu na Ukazatele. Pokud jsou použity Účty, které nejsou mazány, vazba zůstává zachována.		
UCO_5.0	Ukončení platnosti Účtové osnovy	Systém umožňuje uživateli ukončit platnost Účtové osnovy včetně Účtů v ní obsažených (viz OBE_2.0). S ukončením platnosti Účtové osnovy je ukončena platnost Účtům v ní zařazených a současně v odpovídajícím časovém řezu je zrušena vazba na Ukazatele při dodržení podmínek závislosti (viz kapitola 2.3.3 Objekt #ObjektZávislost).	Závazný	1
UCO_5.1	Ukončení platnosti Účtové osnovy systémem	Systém automaticky ukončuje platnost verzi/variantě objektu Účtová osnova, jehož následující verzi/variantě byl uživatelem změněn stav Schválený na Platný (viz OBE_2.1).	Závazný	1
UCO_6.0	Prodloužení platnosti Účtové osnovy	Systém umožňuje uživateli prodloužit platnost Účtové osnovy včetně Účtů v ní obsažených (viz OBE_3.0).	Závazný	1
UCO_7.0	Změna obsahu Účtové osnovy	Systém umožňuje změnit obsah Účtové osnovy (tj. založit nový Účet, ukončit časovou platnost Účtu, smazání Účtu), je-li vytvořena nová verze Účtové osnovy v odpovídajícím časovém řezu a je ve stavu Projektovaný.	Závazný	1
UCO_8.0	Schválení Účtové osnovy	Systém umožňuje uživateli schválit instanci objektu Účtová osnova (viz OBE_4.0).	Závazný	1
UCO_9.0	Zplatnění účtové osnovy	Systém umožňuje uživateli zplatnit instanci objektu Účtová osnova (viz OBE_5.0).	Závazný	1
UCO_10.0	Vrácení Účtové osnovy do	Systém umožňuje převést instanci objektu Účtová osnova ze stavu Schválený do stavu Projektovaný (viz OBE_6.0).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	projekce			
UCO_21.0	Vytvoření účtu	Systém umožňuje uživateli vytvořit Účet v rámci definované Účtové osnovy za splnění byznys podmínek definovaných v kapitole 3.13 Objekt Účet.	Závazný	1
UCO_21.1	Jednoznačnost kódu objektu Účet	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Účet je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Účet, které se vztahují ke shodné verzi/variantě instance Účtové osnovy. Kód objektu lze změnit jen v případě, že existuje pouze první verze, a to ve stavu Projektovaný, a není nikde použitá.	Závazný	1
UCO_22.0	Editace Účtu	Systém umožňuje měnit atributy existujícího Účtu za splnění byznys podmínek uvedených v kapitole 3.13 Objekt Účet. Systém kontroluje, zda prováděné změny jsou v souladu s nastavením systému, které definuje, jak mohou být jednotlivé atributy měněny.	Závazný	1
UCO_23.0	Smazání Účtu	Systém umožňuje smazat Účet ve verzi Účtové osnovy, kde byl vytvořen. Účtová osnova musí být ve stavu Projektovaný. Pokud je Účet použit, dojde ke zrušení vazby tohoto Účtu na Ukazatele.	Závazný	1
UCO_24.0	Ukončení platnosti Účtu	Systém umožňuje ukončit časovou platnost Účtu za splnění byznys podmínek definovaných v kapitole 3.13 Objekt Účet. S ukončením platnosti Účtu je současně v odpovídajícím časovém řezu zrušena vazba daného Účtu na Ukazatele při dodržení podmínek závislosti (viz kapitola 2.3.3 Objekt #ObjektZávislost).	Závazný	1
UCO_25.0	Prodloužení platnosti Účtu	Systém umožňuje prodloužit platnost Účtu, který byl ukončen, za podmínky, že poslední instance Účtové osnovy má platnost_do vyšší nebo rovnu platnost_do Účtu.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
UCO_31.0	Použití Účtové osnovy	Systém umožňuje uživateli zobrazit, které mají definovánu vazbu na Účty z dané Účetní osnovy (viz kapitola 2.4.1 Základní vlastnosti objektů).	Závazný	1
UCO_32.0	Historie Účtové osnovy	Systém umožňuje uživateli zobrazit historii Účtové osnovy (viz KNH_3.0).	Závazný	1
UCO_33.0	Použití Účtu	Systém umožňuje uživateli zobrazit Ukazatele, které mají vazbu na vybraný Účet (viz 2.4.1 Základní vlastnosti objektů).	Závazný	1
UCO_34.0	Historie Účtu	Systém umožňuje uživateli zobrazit historii Účtu (viz KNH_3.0).	Závazný	1
UCO_41.0	Zobrazení vazby na Účet ze vzoru Výkazu	Systém umožňuje uživateli zobrazit v grafické struktuře Výkazu souhrnný přehled k jednotlivým Ukazatelům vazbu na Účet.	Závazný	2

7.15 Datový typ

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DAT_1.0	Zobrazení Datového typu	Systém umožňuje zobrazení náhledu na instance objektů Datový typ tak, že zobrazuje seznam všech definovaných Datových typů a pro uživatelem vybranou instanci objektu Datový typ zobrazí detaily podle kapitoly 3.14 Objekt Datový typ.	Závazný	1
DAT_2.0	Vytvoření Datového typu	Systém umožňuje uživateli vytvořit nový Datový typ (první instanci objektu Datový typ) za splnění byznys podmínek uvedených v kapitole 3.14 Objekt Datový typ. Při vytvoření nového Datového typu systém automaticky vytvoří novou verzi a automaticky tuto verzi založí ve stavu Projektovaný. Datové typy je možné zakládat (při splnění	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		pravidel platných pro ruční vytváření) též hromadně pomocí importu vhodně strukturovaného XML nebo CSV souboru.		
DAT_2.1	Jednoznačnost kódu objektu Datový typ	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Datový typ je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Datový typ. Systém nepovoluje změnit kód objektu na hodnotu již použitou u jiné instance objektu Datový typ.	Závazný	1
DAT_2.2.	Formát kódu Datový typ	Kód instance objektu Datový typ se skládá z prefixu a dalších znaků: • prefix vybírá uživatel nabídkou z listu číselníkových položek, např. n, p, R, viz DAT_2.4, k vybranému prefixu uživatel přiřazuje další alfanumerické znaky nebo podtržítko.	Závazný	1
DAT_2.3	Změna kódu objektu Datový typ	Systém umožňuje uživateli změnit kód instance objektu Datový typ v případě, že existuje pouze první verze, a to ve stavu Projektovaný a není nikde použita. Uživatel kód instance objektu změní jeho přepsáním.	Závazný	1
DAT_2.4	Prefix kódu objektu Datový typ	Systém umožňuje uživateli aktualizovat (zakládat nové, ukončovat platnost a mazat nepoužité položky) číselník prefixů kódů pro objekt Datový typ.	Závazný	1
DAT_3.0	Editace Datového typu	Systém umožňuje měnit atributy existujícího Datového typu (existující instance) za splnění byznys podmínek uvedených v kapitole 3.14 Objekt Datový typ. Před zahájením editace systém umožňuje uživateli se rozhodnout, zda má být založena nová verze/varianta nebo změna bude provedena v rámci existující verze/varianty. Systém kontroluje, zda prováděné změny jsou v souladu s nastavením systému, které definuje, jak mohou být jednotlivé atributy měněny.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DAT_3.1	Změna varianty Datového typu ve verzi	Systém umožňuje uživateli změnit variantu Datového typu ve verzi (viz OBE_7.0).	Závazný	1
DAT_4.0	Smazání Datového typu	Systém umožňuje uživateli smazat Datový typ ve stavu Projektovaný, pokud není nikde použit (viz OBE_1.0).	Závazný	1
DAT_4.1	Smazání Datového typu včetně DDT	Systém umožňuje uživateli smazat Datový typ ve stavu Projektovaný včetně souvisejících Domén datového typu, pokud nejsou nikde použity.	Závazný	1
DAT_5.0	Ukončení platnosti Datového typu	Systém umožňuje uživateli ukončit platnost Datového typu, pokud není nikde použit (viz OBE_2.0).	Závazný	1
DAT_5.1	Ukončení platnosti Datového typu včetně DDT	Systém umožňuje uživateli ukončit platnost Datového typu včetně souvisejících Domén datového typu, pokud nejsou nikde použity.	Závazný	1
DAT_5.2	Ukončení platnosti Datového typu systémem	Systém automaticky ukončuje platnost verzi/variantě objektu Datový typ, jehož následující verzi/variantě byl uživatelem změněn stav Schválený na Platný (viz OBE_2.1).	Závazný	1
DAT_6.0	Prodloužení platnosti Datového typu	Systém umožňuje uživateli prodloužit platnost Datového typu (viz OBE_3.0).	Závazný	1
DAT_7.0	Schválení Datového typu	Systém umožňuje uživateli schválit instanci objektu Datový typ (viz OBE_4.0).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DAT_8.0	Zplatnění Datového typu	Systém umožňuje uživateli zplatnit instanci objektu Datový typ (viz OBE_5.0).	Závazný	1
DAT_9.0	Vrácení Datového typu do projekce	Systém umožňuje převést instanci objektu Datový typ ze stavu Schválený do stavu Projektovaný (viz OBE_6.0).	Závazný	1
DAT_10.0	Použití Datového typu	Systém umožňuje uživateli zobrazit použití Datového typu (viz kapitola 2.4.1 Základní vlastnosti objektů).	Závazný	1
DAT_11.0	Historie Datového typu	Systém umožňuje uživateli zobrazit historii Datového typu (viz KNH_3.0).	Závazný	1

7.16 Doména datového typu

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DDT_1.0	Zobrazení Domény datového typu	Systém umožňuje zobrazení náhledu na instance objektů Doména datového typu tak, že zobrazuje seznam všech definovaných Domén datových typů a pro uživatelem vybranou instanci objektu Doména datového typu zobrazí detaily podle kapitoly 3.15 Objekt Doména datového typu.	Závazný	1
DDT_2.0	Vytvoření Domény datového typu	Systém umožňuje uživateli vytvořit novou Doménu datového typu (první instanci objektu Doména Datového typu), za splnění byznys podmínek uvedených v kapitole 3.15 Objekt Doména datového typu. Při vytvoření nové Domény datového typu systém automaticky vytvoří novou verzi a automaticky tuto verzi založí ve stavu Projektovaný.	Závazný	1
DDT_2.1	Jednoznačnost	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	kódu objektu Doména datového typu	instance objektu Doména datového typu je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Doména datového typu, které se vztahují ke shodné instanci objektu Datový typ.		
DDT_3.0	Editace Domény datového typu	Systém umožňuje měnit atributy existující Domény datového typu (existující instance) za splnění byznys podmínek uvedených v kapitole 3.15 Objekt Doména datového typu. Před zahájením editace systém umožňuje uživateli se rozhodnout, zda má být založena nová verze/varianta nebo změna bude provedena v rámci existující verze/varianty. Systém kontroluje, zda prováděné změny jsou v souladu s nastavením systému, které definuje, jak mohou být jednotlivé atributy měněny.	Závazný	1
DDT_3.1	Změna varianty Domény datového typu ve verzi	Systém umožňuje uživateli změnit variantu Domény datového typu ve verzi (viz OBE_7.0).	Závazný	1
DDT_4.0	Smazání Domény datového typu	Systém umožňuje uživateli smazat Doménu datového typu ve stavu Projektovaný, pokud není nikde použit (viz OBE_1.0).	Závazný	1
DDT_5.0	Ukončení platnosti Domény datového typu	Systém umožňuje uživateli ukončit platnost Domény datového typu, pokud není nikde použit (viz OBE_2.0).	Závazný	1
DDT_5.1	Ukončení platnosti Domény datového typu systémem	Systém automaticky ukončuje platnost verzi/variantě objektu Domény datového typu, jehož následující verzi/variantě byl uživatelem změněn stav Schválený na Platný (viz OBE_2.1).	Závazný	1
DDT_6.0	Prodloužení platnosti Domény	Systém umožňuje uživateli prodloužit platnost Domény datového typu (viz OBE_3.0).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	datového typu			
DDT_7.0	Schválení Domény datového typu	Systém umožňuje uživateli schválit instanci objektu Domény datového typu (viz OBE_4.0).	Závazný	1
DDT_8.0	Zplatnění Domény datového typu	Systém umožňuje uživateli zplatnit instanci objektu Doména datového typu (viz OBE_5.0).	Závazný	1
DDT_9.0	Vrácení Domény datového typu do projekce	Systém umožňuje převést instanci objektu Domény datového typu ze stavu Schválený do stavu Projektovaný (viz OBE_6.0).	Závazný	1
DDT_10.0	Použití Domény datového typu	Systém umožňuje uživateli zobrazit použití Domény datového typu (viz kapitola 2.4.1 Základní vlastnosti objektů).	Závazný	1
DDT_11.0	Historie Domény datového typu	Systém umožňuje uživateli zobrazit historii Domény datového typu (viz KNH_3.0).	Závazný	1

7.17 Ukazatel, Konkretizace ukazatele, Dodatečná konkretizace ukazatele

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
UKZ_1.0	Zobrazení Ukazatele a jeho detailů	Systém umožňuje zobrazení náhledu na instance objektu Ukazatel. Součástí náhledu je jednak zobrazení všech atributů instance objektu Ukazatel a další podrobností, které vyplývají z napojení objektu Ukazatel na ostatní objekty systému. Systém v seznamu Ukazatelů automaticky zobrazuje nejaktuálnější verzi a variantu instance objektu Ukazatel, nicméně umožňuje získat informace	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie

		i o všech jeho předcházejících verzích/variantách.
		
		Pro vybraný Ukazatel tak systém v rámci náhledu prezentuje zejména tyto informace:
		
		• zobrazení kompletní historie všech verzí a variant, včetně vymezení časové oblasti každé verze/varianty, kterými daný Ukazatel prošel během svého životního cyklu,
		• zobrazení kompletní sady stavů, kterými prošla ta která verze/varianta Ukazatele,
		• zobrazení informací o Účtech, které jsou k Ukazateli přiřazeny. Tato vazba je v objektovém modelu podchycena asociační třídou Zařazení účtu k ukazateli; rozsah zobrazených informací musí být v souladu s popisem přiřazení Účtu k Ukazateli. Tato pravidla jsou popsána v kapitole 3.13 Objekt Účet,
		• zobrazení informací o všech Datových oblastech, kde je daný Ukazatel použit (asociační třída Ukazatel v DO). Pro každou kombinaci Ukazatel/Datová oblast zobrazeno zejména:
		  • informace o kódu a názvu Datové oblasti, verzi/variantě, ke které je připojena daná verze/varianta Ukazatele a verze/varianty Výkazu, do kterého je daná Datová oblast zařazena, informaci o pořadí daného Ukazatele v rámci dané Datové oblasti a informaci od nadřízeném Ukazateli (je-li definován),
		  • informace, jakým způsobem je Ukazatel do Datové oblasti zařazen. Jedná se o zobrazení informace o tom, zda je daná instance objektu Konkretizace ukazatele napojena na instanci objektu Datový Typ, Doména (Doména číselníku nebo Doména datového typu), Položka (Položka číselníku nebo Položka hierarchie) nebo Hierarchie číselníku. Instance		

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		objektu Konkretizace ukazatele může být napojena na právě jednu instanci právě jednoho z výše uvedených objektů; XOR, • informace o tom, zda a jak je Ukazatel dodatečně konkretizován (asociační třída Dodatečná konkretizace ukazatele), podrobně je tato vazba a celý princip dodatečné konkretizace popsán v kapitole 3.16.1 Objekt Dodatečná konkretizace Ukazatele, • informace o tom, s jakými Účty je Ukazatel do dané Datové oblasti zařazen (objekt Ukazatel v DO, atribut Detailní rozpis účtů).		
UKZ_2.0	Vytvoření Ukazatele	Systém umožňuje vytvořit nový Ukazatel (instanci objektu Ukazatel) za splnění byznys podmínek uvedených v kapitole 3.16 Objekt Ukazatel. V rámci definice nového Ukazatele je možné definovat vazbu Ukazatele na Účet. Jedná se o vytvoření instance(í) objektu Zařazení účtu k ukazateli. Účty je možno k Ukazateli přiřazovat za splnění pravidel, která jsou popsána v kapitole 3.13 Objekt Účet. V rámci definice ukazatele **neprobíhá** vytváření instance objektu Ukazatel v DO. Tato instance vzniká v okamžiku zařazování Ukazatele do Datové oblasti a je popsána ve funkčním požadavku UKZ_6.0. Při vytvoření nového Ukazatele systém automaticky založí novou verzi Ukazatele a automaticky tuto verzi založí ve stavu Projektovaný. Podrobně je popis životního cyklu instancí popsán v kapitole 2.2 Sledování historie instancí objektů metapopisu.	Závazný	1
UKZ_2.1	Jednoznačnost kódu objektu Ukazatel	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Ukazatel je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Ukazatel. Systém nepovoluje změnit kód objektu na hodnotu již použitou u jiné instance	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		objektu Ukazatel.		
UKZ_2.2.	Formát kódu Ukazatele	Kód instance objektu Ukazatel se skládá z prefixu a pořadového čísla: • prefix vybírá uživatel nabídkou z listu číselníkových položek, např. EPR, SOLV, viz UKZ_2.4, • k vybranému prefixu systém nabízí číselnou 4-místnou hodnotu, tj. například EPR0001 pro první Ukazatel odvozený od prefixu EPR. Pro další Ukazatele systém nabízí číselnou hodnotu vyšší o 1 (např. EPR0002). • Pro 2. Úroveň Ukazatele se kód instance objektu skládá z kódu Ukazatele 1. Úrovně, na který má jedinečnou vazbu, a dále ze sufixu. Kód Ukazatele 2. Úrovně vypadá např. EAN0003_001.	Závazný	1
UKZ_2.3	Změna kódu objektu Ukazatel	Systém umožňuje uživateli změnit kód instance objektu Ukazatel v případě, že existuje pouze první verze, a to ve stavu Projektovaný a není nikde použita. Uživatel kód instance objektu změní jeho přepsáním.	Závazný	1
UKZ_2.4	Prefix kódu objektu Ukazatel	Systém umožňuje uživateli aktualizovat (zakládat nové, ukončovat platnost a mazat nepoužité položky) číselník prefixů kódů pro objekt Ukazatel.	Závazný	1
UKZ_3.0	Editace Ukazatele	Systém umožňuje měnit atributy existujícího Ukazatele (existující instance objektu Ukazatel) za splnění byznys podmínek uvedených v kapitole 3.16 Objekt Ukazatel. Před zahájením editace vybraného Ukazatele systém umožňuje uživateli se rozhodnout, zda má být založena nová verze/varianta nebo zda má být provedena změna v rámci existující verze/varianty. Po provedení změn systém kontroluje, zda jsou tyto změny v souladu s nastavením systému, které definuje, jak mohou být jednotlivé atributy měněny. Podrobně jsou pravidla pro měnění jednotlivých atributů popsány v kapitole 2.3 Vazby	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		mezi jednotlivými objekty.		
UKZ_4.0	Smazání Ukazatele	Systém umožňuje smazat existující Ukazatel pouze v případě, že tento Ukazatel není připojen k žádné Datové oblasti (pro daný Ukazatel neexistuje žádná související instance objektu Ukazatel v Datové oblasti) a v souladu s funkčním požadavkem OBE_1.0. Existence vazeb instance objektu Ukazatel na instance objektu Účet (asociační třída Zařazení účtu k ukazateli) nemá na rozhodnutí o tom, zda je možno smazání provést nebo ne, žádný vliv. V případě, že jsou splněny všechny podmínky pro smazání Ukazatele a uživatel tuto akci provede, pak dochází ke smazání všech souvisejících instancí objektu Zařazení účtu k ukazateli. Samotné instance objektu Účet nejsou smazáním Ukazatele nijak dotčeny.	Závazný	1
UKZ_5.0	Ukončení platnosti Ukazatele	Systém umožňuje ukončit časovou platnost vybrané verze Ukazatele v souladu s OBE_2.0. Ukončením platnosti se rozumí změna data času platnosti dané verze (atribut platnost_do). Podrobně je postup ukončování platnosti verze popsán v kapitole 2.2.6 Přístup „Sledování historie – časová platnost + stavy“.	Závazný	1
UKZ_6.0	Vložení Ukazatele do Datové oblasti	Systém umožňuje vložení Ukazatele do Datové oblasti (instance objektu Ukazatel v DO), a to tak, že vybrané Datové oblasti přiřadí vybraný Ukazatel. V rámci přiřazení Ukazatele do Datové oblasti je nutné definovat tzv. **pořadí Ukazatele v Datové oblasti** a **osu Datové oblasti**, do které bude Ukazatel umístěn. Tyto informace jsou důležité z hlediska následného procesu generování Údaje (viz kapitola 7.19 Údaj, UDJ_1.0 a další tamtéž). Systém umožňuje zařazovat Ukazatele do Datové oblasti hierarchicky. Zachycení vazby nadřízenosti/podřízenosti jednotlivých ukazatelů	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		zajišťuje vazba nadřízený Ukazatel v Datové oblasti. Pro zařazení Ukazatele do Datové oblasti platí tato pravidla: • hierarchie Ukazatelů je nepovinná. To znamená, že uživatel může zařadit všechny Ukazatele do Datové oblasti tak, že všechny budou na nejvyšší úrovni a žádná hierarchie tak nevznikne. V rámci jedné Datové oblasti může být na nejvyšší úrovni N Ukazatelů, • uživatel může při zařazování Ukazatele do Datové oblasti určit, že daný Ukazatel se má stát nadřízeným nebo podřízeným již zařazenému Ukazateli, • pokud má Ukazatel v Datové oblasti vazbu na nadřízený Ukazatel, má právě jednu takovou vazbu (jeden Ukazatel v Datové oblasti nemůže mít dva a více přímých nadřízených Ukazatelů), • pokud uživatel vytváří v Datové oblasti hierarchii Ukazatelů, pak musí u každého Ukazatele, který má nějaké podřízené Ukazatele určit, zda je výčet podřízených položek úplný. Systém tuto informaci uchová takto: - nastaví hodnotu atributu „suma“ na „ano“ v případě, že uživatel určí, že daný Ukazatel obsahuje úplný výčet podřízených Ukazatelů, - nastaví hodnotu atributu „suma“ na „ne“, v případě, že uživatel určí, že výčet podřízených Ukazatelů není úplný. Atribut „suma“, resp. jeho hodnota má následně vliv na tzv. automaticky generované kontroly (viz JVK_2.0). Akci je možno provést pouze v případě, že existují instance objektů Datová oblast a Ukazatel. Vložit Ukazatele do Datové oblasti je možno pouze v případě, že Datová oblast, do které má být Ukazatel vložen, je ve stavu Projektovaný.		

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Do projektované Datové oblasti lze zařadit pouze ten Ukazatel, jehož časová platnost odpovídá časové platnosti Datové oblasti, kam je zařazován; nezáleží na stavu Ukazatele (to znamená, že do Datové oblasti lze zařadit Ukazatel, který je ve stavu Projektovaný, Schválený, Platný). V rámci procesu zařazování Ukazatele do Datové oblasti musí uživatel definovat další informace (viz kapitola 3.16 Objekt Ukazatel).		
UKZ_6.1.	Vytvoření Dodatečné konkretizace ukazatele	Systém umožňuje ke každému Ukazateli definovat tzv. dodatečnou konkretizaci. Jedná se o provázání existující instance objektu Ukazatel s jednou nebo více instancemi objektu Parametr přes asociační třídu Dodatečná konkretizace ukazatele. Pokud je Ukazatel zařazen do Datové oblasti, pak vytvořit novou instanci objektu Dodatečná konkretizace ukazatele je možno pouze v případě, že Datová oblast, ve které je Ukazatel zařazen, je ve stavu Projektovaný. Systém zajišťuje vytvoření Dodatečné konkretizace ukazatele v souladu se všemi byznys podmínkami popsanými v kapitole 3.16.1 Objekt Dodatečná konkretizace Ukazatele.	Závazný	1
UKZ_7.0	Změna Ukazatele v Datové oblasti	Systém umožňuje změnit existující Ukazatel v Datové oblasti pouze v případě, že tato Datová oblast, je ve stavu Projektovaný. Změnou Ukazatele v Datové oblasti se myslí: • **změna pořadí Ukazatele v Datové oblasti:** uživatel může změnit pořadí jednotlivých Ukazatelů, • **změna nadřízeného Ukazatele:** uživatel může změnit nadřízený Ukazatel v Datové oblasti, jedná se o tyto akce: ○ uživatel smaže odkaz na nadřízený Ukazatel v Datové oblasti, tzn., že se z takového Ukazatele stane Ukazatel nejvyšší	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		úrovně v rámci Datové oblasti, ○ uživatel přidá informaci o nadřízeném Ukazateli tomu Ukazateli, který doposud žádný nadřízený Ukazatel v rámci Datové oblasti neměl. Z Ukazatele nejvyšší úrovně se stane podřízený Ukazatel, ○ uživatel změní informaci o nadřízeném Ukazateli, tzn., že se daný Ukazatel přesune v rámci Datové oblasti pod jiný nadřízený Ukazatel. Pokud má daný Ukazatel nějaké podřízené Ukazatele jsou i tyto přesunuty pod jiný nadřízený Ukazatel v Datové oblasti, • **změna/odstranění Dodatečné konkretizace ukazatele:** změna/odstranění Dodatečné konkretizace se řídí pravidly uvedené v kapitole 3.16.1 Objekt Dodatečná konkretizace Ukazatele, • **změna ukazatele:** uživatel určí, že Ukazatel již dále nemá být připojen na jednu a tu samou Datovou oblast a místo něj má být použit jiný Ukazatel. Pokud je měněný Ukazatel součástí hierarchické struktury Ukazatelů v Datové oblasti, systém zajišťuje, že nedojde k rozpadu této hierarchie. Výsledkem celé operace je tak výměna jednoho Ukazatele za jiný.		
UKZ_8.0	Odebrání Ukazatele z Datové oblasti	Systém umožňuje odebrat Ukazatel v Datové oblasti pouze v případě, že daná Datová oblast je ve stavu Projektovaný. Pokud je odebíraný Ukazatel nadřízený jinému Ukazateli v dané Datové oblasti (vazba „nadřízený Ukazatel v Datové oblasti“), pak systém upozorní na tuto skutečnost uživatele a nabídne tyto možnosti: • odebrání Ukazatele z Datové oblasti, včetně odebrání všech jeho podřízených Ukazatelů v Datové oblasti v celé hloubce stromu, • přesunutí přímých podřízených Ukazatelů v Datové oblasti na	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		nejvyšší úroveň, • přesunutí přímých podřízených položek pod jiný Ukazatel.		
UKZ_11.0	Změna stavu Ukazatele ze Schválený na Projektovaný	Systém umožňuje provést návrat ke stavu Projektovaný ze stavu Schválený v souladu s pravidly definovanými v OBE_6.0 u každé instance objektu Ukazatel.	Závazný	1
UKZ_12.0	Zplatnění nové verze/varianty Ukazatele	Systém umožňuje zplatnění nové verze Ukazatele v souladu s pravidly definovanými v OBE_5.0	Závazný	1
UKZ_13.0	Ukončení platnosti instance objektu Ukazatel	Systém umožňuje ukončení platnosti instance objektu Ukazatel v souladu s FP OBE_2.1.	Závazný	1
UKZ_14.0	Schválení instance objektu Ukazatel	Systém umožňuje schválit instanci objektu Ukazatel v souladu s FP OBE_4.0	Závazný	1
UKZ_15.0	Hromadné vytvoření instancí objektu Ukazatel	Systém umožňuje hromadně vytvořit instance objektu Ukazatel (za splnění všech omezujících podmínek definovaných pro individuální vytvoření instance objektu Ukazatel) v souladu s procesem Tvorba objektů popisujících údaje, viz kapitola 5.4 Proces tvorby objektů popisujících údaje), tj. označení oblasti, která obsahuje Ukazatele a následného předvyplnění (název Ukazatele) formuláře pro hromadné vytvoření Ukazatele, kde bude moci uživatel doplnit další atributy Ukazatele.	Závazný	3

7.18 Parametr, Konkretizovaný parametr

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
PAR_1.0	Zobrazení Parametru a jeho detailů	Systém umožňuje zobrazení náhledu na instance objektu Parametr. Součástí náhledu je jednak zobrazení všech atributů instance objektu Parametr a další podrobností, které vyplývají z napojení objektu Parametr na ostatní objekty systému. Systém v seznamu Parametrů automaticky zobrazuje nejaktuálnější verzi/variantu instance objektu Parametr, nicméně umožňuje získat informace i o všech jeho předcházejících verzích/variantách. Pro vybraný Parametr tak systém v rámci náhledu prezentuje zejména tyto informace: • zobrazení kompletní historie všech verzí/variant, včetně vymezení časové platnosti každé verze/varianty, kterými daný Parametr (konkrétní instance objektu Parametr) prošel během svého životního cyklu, • zobrazení kompletní sady stavů, kterými prošla ta která verze/varianta Parametru, • zobrazení informací o typu Parametru, tedy o tom, zda je Parametr napojen na Číselník nebo Datový typ. Podrobněji o napojení Parametru na Číselník a Datový typ pojednává kapitola 3.17 Objekt Parametr, • zobrazení informací o všech Datových oblastech, kde je daný Parametr použit (asociační třída Konkretizovaný parametr). Pro každou kombinaci Parametr/Datová oblast zobrazeno zejména: • informace o kódu a názvu Datové oblasti, verzi/variantě, ke které je připojena daná verze/varianta Parametru, a verze/varianta Výkazu, do kterého je daná Datová oblast zařazena, informace o ose, na kterou je Parametr v rámci dané	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Datové oblasti připojen, a informaci o pořadí daného Parametru v rámci dané osy. • informace, jakým způsobem je Parametr do Datové oblasti zařazen. Jedná se o zobrazení informace o tom, zda je daná instance objektu Konkretizovaný parametr napojena na instanci objektu Datový Typ, Doména (Doména číselníku nebo Doména datového typu), Položka (Položka číselníku nebo Položka hierarchie) nebo Hierarchie číselníku. Instance objektu Konkretizovaný parametr může být napojena na právě jednu instanci právě jednoho z výše uvedených objektů; XOR, • zobrazení informace o tom, jaké Ukazatele vybraný Parametr dodatečně konkretizuje (asociační třída Dodatečná konkretizace ukazatele). Podrobně je tato vazba a celý princip dodatečné konkretizace popsán v kapitole 3.16.1 Objekt Dodatečná konkretizace Ukazatele, • zobrazení seznamu všech Vykazovaných parametrů (objekt Vykazovaný parametr), které se váží k instanci objektu Parametr.		
PAR_2.0	Vytvoření Parametru	Systém umožňuje vytvořit nový Parametr (instanci objektu Parametr) za splnění byznys podmínek uvedených v kapitole 3.17 Objekt Parametr. V rámci definice nového Parametru je nutno určit, jakého typu Parametr bude. Uživatel zajistí, že Parametr bude buď typu „číselník“ nebo „datový typ“, a to tak, že vytvářený Parametr napojí na existující instanci jednoho nebo druhého objektu (XOR; buď Číselník nebo Datový typ, nikdy obě možnosti). V rámci definice Ukazatele **neprobíhá** vytváření instance objektu Konkretizovaný parametr ani instance objektu Dodatečná konkretizace ukazatele. Tyto instance vznikají v okamžiku zařazování Parametru do	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Datové oblasti, resp. procesu dodatečné konkretizace ukazatele a jsou popsány ve funkčních požadavcích PAR_6.0 (konkretizace Parametru), resp. UKZ_5.1 (Dodatečná konkretizace ukazatele). Při vytvoření nového Parametru systém automaticky založí novou verzi Ukazatele a automaticky tuto verzi založí ve stavu Projektovaný. Podrobně je popis životního cyklu instancí popsán v kapitole 2.2 Sledování historie instancí objektů.		
PAR_2.1	Jednoznačnost kódu objektu Parametr	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Parametr je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Parametr. Systém nepovoluje změnit kód objektu na hodnotu již použitou u jiné instance objektu Parametr.	Závazný	1
PAR_2.2.	Formát kódu Parametru	Kód instance objektu Parametr se skládá z prefixu a pořadového čísla: • prefix vybírá uživatel nabídkou z listu číselníkových položek, např. P, I, viz PAR_2.4, k vybranému prefixu systém nabízí číselnou 4-místnou hodnotu, tj. například T0001 pro první Parametr odvozený od prefixu T. Pro další Parametry systém nabízí číselnou hodnotu vyšší o 1 (např. T0002).	Závazný	1
PAR_2.3	Změna kódu objektu Parametr	Systém umožňuje uživateli změnit kód instance objektu Parametr v případě, že existuje pouze první verze, a to ve stavu Projektovaný a není nikde použita. Uživatel kód instance objektu změní jeho přepsáním.	Závazný	1
PAR_2.4	Prefix kódu objektu Parametr	Systém umožňuje uživateli aktualizovat (zakládat nové, ukončovat platnost a mazat nepoužité položky) číselník prefixů kódů pro objekt Parametr.	Závazný	1
PAR_3.0	Změna Parametru	Systém umožňuje měnit atributy existujícího Parametru (existující	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		instance objektu Parametr) za splnění byznys podmínek uvedených v kapitole 3.17 Objekt Parametr. Před zahájením editace vybraného Parametru systém umožňuje uživateli se rozhodnout, zda má být založena nová verze, varianta nebo zda má být provedena změna v rámci existující verze/varianty. Po provedení změn systém kontroluje, zda jsou tyto změny v souladu s nastavením systému, které definuje, jak mohou být jednotlivé atributy měněny. Podrobně jsou pravidla pro měnění jednotlivých atributů popsány v kapitole 2.3 Vazby mezi jednotlivými objekty.		
PAR_4.0	Smazání Parametru	Systém umožňuje smazat existující instanci objektu Parametr pouze za souladu s podmínkami uvedenými v OBE_1.0, tedy v případě, že tento Parametr není připojen k žádné Datové oblasti (pro daný Parametr neexistuje žádná související instance objektu Konkretizovaný Parametr) a zároveň v případě, že se tento Parametr nepodílí na dodatečné konkretizaci Ukazatele (pro daný Parametr neexistuje žádná související instance objektu Dodatečná konkretizace ukazatele).	Závazný	1
PAR_5.0	Ukončení platnosti Parametru	Systém umožňuje ukončit časovou platnost vybrané verze/varianty Parametru pouze za souladu s podmínkami uvedenými v OBE_2.0. Ukončením platnosti se rozumí změna data času platnosti dané verze (atribut platnost_do). Postup ukončování platnosti verze/varianty popsán v kapitole 2.2.6 Přístup „Sledování historie – časová platnost + stavy“.	Závazný	1
PAR_6.0	Vytvoření Konkretizovaného parametru	Systém umožňuje zařadit Parametr do Datové oblasti a vytvořit tak nový Konkretizovaný parametr (vzniká nová instance objektu Konkretizovaný parametr) v souladu s podmínkami uvedenými v kapitole 3.17 Objekt Parametr.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		V rámci vzniku nového Konkretizovaného parametru je třeba: • definovat, jakým způsobem bude Parametr v Datové oblasti konkretizován (konkretizovat Parametr je možno právě jednou z možností Položka číselníku/hierarchie, Doména číselníku/datového typu, Datový typ nebo Hierarchie číselníku), • určit, na jakou osu Datové oblasti (horizontální osa X/vertikální osa Y/karta osa Z) bude Parametr zařazen (určení osy je nutné jen pro Parametry, které tvoří dimenzi; pro nedimenzionální Parametry není nutné pořadí určovat), • určit pořadí, v jakém bude Parametr na danou osu zařazen (určení pořadí je nutné jen pro Parametry, které tvoří dimenzi), • určit, na jakou kartu Datové oblasti bude Parametr zařazen (pouze v případě, že se jedná o kartotékovou datovou oblast). Přiřazování Parametru do Datové oblasti je možno provést pouze v případě, že se Výkaz, do jehož Datové oblasti je Parametr zařazován, nachází ve stavu Projektovaný.		
PAR_7.0	Změna Konkretizovaného parametru	Systém umožňuje změnit existující Konkretizovaný parametr pouze v případě, že Datová oblast, ke které je existující Parametr přes objekt Konkretizovaný parametr připojen, je ve stavu Projektovaný. Změnou Konkretizovaného parametr se myslí: • **změna typu konkretizace:** uživatel určí, že daný Parametr je stále napojen na jednu a tu samou Datovou oblast, ale změní se jeho konkretizace, tedy určení toho, zda je napojen na Položku číselníku/hierarchie, Doménu číselníku/datového typu, Datový typ nebo Hierarchii (existující instance objektu Konkretizovaný parametr nezaniká),	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		• **změna osy:** uživatel určí, že má být Parametr zařazen na jinou osu Datové oblasti, než je aktuálně zařazen, • **změna pořadí:** uživatel určí, že Parametr má být do Datové oblasti zařazen v jiném pořadí než je aktuálně zařazen, • **změna Parametru:** uživatel určí, že Parametr již dále nemá být připojen na jednu a tu samou Datovou oblast a místo něj má být použit jiný Parametr; dochází tedy k odstranění vazby mezi Datovou oblastí a Parametrem (instance objektu Konkretizovaný parametr zaniká) a tato vazba je nahrazena novou vazbou na jiný Parametr (vzniká nová instance objektu Konkretizovaný parametr).		
PAR_8.0	Smazání Konkretizovaného parametru	Systém umožňuje smazat existující Konkretizovaný parametr pouze v případě, že Datová oblast, ke které je existující Parametr přes objekt Konkretizovaný parametr připojen, je ve stavu Projektovaný. Smazání existující instance objektu Konkretizovaný parametr nemá žádný vliv na související instance objektu Datová oblast a Parametr; ty zůstanou zachovány beze změny.	Závazný	1
PAR_9.0	Změna stavu Parametru ze Schválený na Projektovaný	Systém umožňuje provést návrat ke stavu Projektovaný ze stavu Schválený v souladu s pravidly definovanými v OBE_6.0 u každé instance objektu Parametr.	Závazný	1
PAR_10.0	Zplatnění nové verze/varianty Parametru	Systém umožňuje zplatnění nové verze/varianty Parametru v souladu s pravidly definovanými v OBE_5.0.	Závazný	1
PAR_11.0	Ukončení platnosti instance objektu Parametr	Systém umožňuje ukončení platnosti instance objektu Parametr v souladu s funkčním požadavkem OBE_2.1.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
PAR_12.0	Schválení instance objektu Parametr	Systém umožňuje schválit instanci objektu Parametr v souladu s funkčním požadavkem OBE_4.0	Závazný	1

7.19 Údaj

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
UDJ_1.0	Vytvoření Údaje – generování systémem	Systém umožňuje vytvořit v rámci projektované Datové oblasti nový Údaj podle následujícího postupu: • uživatel přiřadí do Datové oblasti minimálně jeden Ukazatel. Přiřazení Ukazatele do Datové oblasti je popsáno v UKZ_2.0. (• uživatel přiřadí do Datové oblasti minimálně jeden Parametr. Přiřazení Parametru do Datové oblasti je popsáno v PAR_2.0, • uživatel použije akci „hromadné generování údajů“. Systém: o požádá uživatele o informaci, jakou má nastavit údajům citlivost. Uživatel vybere právě jednu citlivost, ze seznamu možných hodnot (viz kapitola 3.18.3 Atributy objektu Údaj), tato citlivost pak bude nastavena všem Údajům, které budou v dalším kroku vygenerovány. Uživatel ji pak bude moci individuálně upravit (viz UDJ_2.0), o zkontroluje, zda v Datové oblasti již existují nějaké Údaje. Pokud ano, pak je všechny smaže, o vygeneruje sadu nových instancí objektů Údaj podle algoritmu popsaného v kapitole 3.18.1 Vznik Údaje – statická Datová oblast v rámci procesu generování. Systém kontroluje, zda v daném Výkazu v jiné Datové oblasti již neexistuje Údaj,	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		který by byl stejně popsán (je mu přiřazen stejný ukazatel a stejná sada Konkretizovaných parametrů. V případě, že to tak bude, pak systém: 1) zjistí identifikátor Údaje, který již existuje v jiné Datové oblasti a je definován stejně jako nově vznikající Údaje, 2) vygeneruje nový Údaj se stejnou definicí jako má již existující Údaj a jako hodnotu odvozeného atributu „zobrazovaný údaj“ použije identifikátor získaný v bodě 1). Tím dojde k provázání obou Údajů a nově vzniklý Údaj nebude nutno ze strany Osob v této Datové oblasti vykazovat.		
UDJ_2.0	Změna existujícího Údaje	Systém umožňuje uživateli pro každý existující Údaj změnit hodnotu atributu „vykazovat“. Změna atributu je možná pouze v případě, že se Údaj nachází v Datové oblasti, která je ve stavu Projektovaný. Standardně je při generování Údaje nastaven tento atribut na hodnotu „ano“ (pouze v případě, že je Údaj shledán duplicitním, je nastaven na „ne“). V případě, že uživatel nastaví hodnotu atributu na „ne“, znamená to, že daný Údaj nemá být předmětem vykazování.	Závazný	1
UDJ_2.1	Smazání údaje	Systém neumožňuje smazat žádný vygenerovaný Údaj. Pokud údaj nemá být použit pro vykazování, je nutno provést jeho editaci a nastavit hodnotu atributu „vykazovat“ na „ne“ (viz UDJ_2.0).	Závazný	1
UDJ_2.2	Změna citlivosti Údaje	Systém umožňuje změnit citlivost Údaje tak, že vybere z číselníku jinou hodnotu citlivosti daného Údaje. Citlivost Údaje je možné měnit z Datové oblasti, která je ve stavu Projektovaný, Schválený nebo Platný.	Závazný	1
UDJ_3.0	Údaj - zobrazení	Systém umožňuje k vybrané Datové oblasti zobrazit seznam všech Údajů, které jsou v rámci ní vygenerovány. Toto zobrazení je možné bez ohledu	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		na to, v jakém stavu se Datová oblast nachází. Zobrazení Údajů v Datové oblasti probíhá následovně: 1) systém nejdříve vykreslí strukturu Datové oblasti. Struktura Datové oblasti je vykreslena na základě souvisejících instancí objektů Ukazatel v Datové oblasti a Konkretizovaný Parametr tak, že se zohlední atributy „pořadí“ a „osa“ (v případě Parametru), 2) výsledkem operace z bodu 1) je struktura (grid) odpovídající souřadnicím Údajů, neboli jednotlivé buňky Datové oblasti, které představují jednotlivé Údaje, 3) systém zobrazí jako neaktivní (například šedou barvou), ty buňky, které představují Údaj, který je označen jako „nevykazovaný“, Systém umožňuje uživateli označit jakoukoli buňku výsledného gridu a zobrazit její tzv. diagnostiku. Pod pojmem diagnostika buňky rozumíme: • informaci o Ukazateli, který je danému Údaji přiřazen, • informaci o seznamu všech společných Parametrů, které jsou k Údaji přiřazeny (Parametry společné pro všechny Údaje v Datové oblasti), • informaci o seznamu všech nespolečných Parametrů (Parametry přiřazené pouze tomuto konkrétnímu Údaji).		
UDJ_4.0	Výběr jednoho z duplicitních údajů jako vykazovaného	V případě, že v rámci jednoho Výkazu existuje více Údajů, které jsou stejně popsány (tzv. duplicitní Údaje), pak systém umožňuje uživateli zvolit, který z těchto duplicitních údajů (v které Datové oblasti) má být vykazovaný. Musí být splněno pravidlo, že právě jeden z těchto duplicitních Údajů má být vykazovaný (hodnota atributu „vykazovat“ je nastavena na „ano“) a že všechny ostatní Údaje odkazují na tento jeden vykazovaný Údaj (odvozený atribut zobrazovaný údaj).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
UDJ_5.0	Dynamické atributy údaje	Systém umožňuje dle kapitoly 3.18.4 Dynamické atributy Údaje specifikovat dynamické atributy údaje.	Závazný	2
UDJ_6.0	Údaj – atribut popisek údaje	Definuje popisek, který bude dále možné využít jako „label“. Možné využít pouze pro typ Datové oblasti – volná.	Závazný	2

7.20 Jednovýkazová kontrola (JVK)

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
JVK_1.0	JVK – sémantický jazyk JVK	Systém umožňuje uživateli zapsat JVK prostřednictvím sémantického jazyka pro tvorbu kontrol (viz kapitola 3.19.3.1 Kontroly vytvořené sémantickým jazykem).	Závazný	1
JVK_1.1	JVK – algoritmická	Systém umožňuje uživateli vytvořit algoritmickou JVK (viz kapitola 3.19.3.2 Algoritmické kontroly) výběrem algoritmu JVK ze seznamu (viz JVK_9.2) a jeho aplikací na konkrétní Údaj nebo Údaje.	Závazný	2
JVK_1.2	JVK – sledování historie	Systém u každé instance objektu JVK sleduje její historii podle kapitoly 3.19.3 Objekt Jednovýkazová kontrola (JVK).	Závazný	1
JVK_1.3	JVK – vytvoření algoritmu dodavatelem	Systém je dodán včetně algoritmů pro JVK uvedených v kapitole 8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systém SDAT.	Závazný	2
JVK_2.0	JVK – generované systémem	Systém umožňuje uživateli spustit generování JVK na základě vazeb použitých instancí objektů popisujících Údaje v rámci jedné instance objektu Výkaz (viz kapitola 3.19.3 Objekt Jednovýkazová kontrola (JVK)) v souladu s postupem definovaným v kapitole 5.5.2.1 Věcné kontroly generované automaticky systémem.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		V případě, že již byly nad instancí objektu Výkaz vygenerovány JVK, uživatel opětovným spuštěním generování Jednovýkazových kontrol, všechny dříve vygenerované JVK smaže (viz JVK_8.3), nebo jim ukončí platnost (viz JVK_8.1). Systém defaultně generuje JVK pro všechny Údaje instance objektu Výkaz, pro které lze tyto kontroly vygenerovat. Systém generuje kontroly pouze v sémantickém jazyce (viz JVK_1.0) a pouze pro instanci objektu Výkaz, jenž se nachází ve stavu Projektovaný.		
JVK_2.1	JVK – generované systémem – označení Údajů pro negenerování JVK	Systém umožňuje uživateli označit Údaje v rámci jedné instance objektu Výkaz, pro které nechce vygenerovat JVK (viz kapitola 5.5.2.1 Věcné kontroly generované automaticky systémem).	Závazný	1
JVK_2.2	JVK – generované systémem – stanovení odchylky JVK systémem	Systém pro JVK vytvořené podle JVK_2.0 stanovuje odchylku na základě pravidel popsaných v kapitole 3.19.6 Odchylka v sémantických kontrolách.	Závazný	1
JVK_2.3	JVK – generované systémem – atributy nastavené systémem	Systém nastavuje JVK vytvořeným podle JVK_2.0 následující atributy: • interní identifikátor objektu, • kód objektu, • název objektu, • autor objektu (přihlášený uživatel), • datum vytvoření (aktuální datum),	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		• kdo aktualizoval (přihlášený uživatel), • datum a čas aktualizace (aktuální datum), • platnost_od (platnost_od výkazu), • platnost_do (platnost_do výkazu), • úroveň závažnosti (závažná chyba), • sémantický tvar vzorce kontroly (viz JVK_2.0), • uživatelský tvar kontroly (viz JVK_10.1), • automatická		
JVK_2.4	JVK – generované systémem – atributy zadávané uživatelem	Systém umožňuje uživateli vyplnit atributy: • popis objektu, • poznámka. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
JVK_2.5	JVK – generované systémem – atributy (zadané systémem) měněné uživatelem	Systém umožňuje uživateli měnit následující atributy JVK (viz kapitola 2.3.6 Rámcové vymezení závislosti objektů) podle JVK_2.3: • název objektu, • úroveň závažnosti. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
JVK_2.6	JVK – generované systémem – změna odchylky uživatelem	Systém umožňuje uživateli změnit odchylku JVK stanovenou systémem podle JVK_2.2 za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
JVK_3.0	JVK – vytvořena	Systém umožňuje uživateli ručně vytvořit JVK, která je popsána	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	uživatelem	základními atributy (viz kapitola 3.19.1 Atributy objektu Kontrola). Systém umožňuje uživateli vytvořit JVK v sémantickém jazyce (viz JVK_1.0) i algoritmickým zápisem (viz JVK_1.1), a to pouze pro instanci objektu Výkaz, jenž se nachází ve stavu Projektovaný (viz kapitola 5.5.2.2 Věcné kontroly vytvářené ručně uživatelem).		
JVK_3.1	JVK – vytvořena uživatelem – sémantický zápis	Systém umožňuje uživateli vytvořit JVK podle JVK_3.0 přímým zápisem sémantického tvaru kontroly.	Závazný	1
JVK_3.2	JVK – vytvořena uživatelem – zápis průvodcem	Systém umožňuje uživateli vytvořit JVK pomocí průvodce (viz kapitola 5.5.2.2 Věcné kontroly vytvářené ručně uživatelem).	Závazný	1
JVK_3.3	JVK – vytvořena uživatelem – stanovení odchylky JVK	Systém umožňuje uživateli stanovit odchylku JVK přímým zápisem do sémantického tvaru (viz JVK_3.1) nebo pomocí průvodce (viz JVK_3.2).	Závazný	1
JVK_3.4	JVK – vytvořena uživatelem – atributy nastavené systémem	Systém nastavuje JVK vytvořeným podle JVK_3.0 následující atributy: • interní identifikátor objektu, • kód objektu, • autor objektu (přihlášený uživatel), • datum vytvoření (aktuální datum), • kdo aktualizoval (přihlášený uživatel), • datum a čas aktualizace (aktuální datum), • platnost_od (platnost_od výkazu), • platnost_do (platnost_do výkazu), • úroveň závažnosti (závažná chyba), • uživatelský tvar kontroly (viz JVK_10.1).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
JVK_3.5	JVK – vytvořena uživatelem – atributy zadávané uživatelem	Systém umožňuje uživateli vyplnit atributy: • název objektu, • popis objektu, • poznámka, • sémantický tvar vzorce kontroly. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
JVK_3.6	JVK – vytvořena uživatelem – atributy měněné uživatelem	Systém umožňuje uživateli měnit následující atributy JVK podle JVK_3.4: • kód objektu, • úroveň závažnosti. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
JVK_4.0	JVK – vytvořené replikací	Systém vytváří JVK pro instance objektů Datová oblast, které vznikly podle DOB_1.1 a DOB_1.2. Takto vytvořené JVK mají stejné atributy (vyjma atributu interní identifikátor) jako JVK, z nichž byly vytvořeny.	Závazný	1
JVK_5.0	JVK – vytvoření verze/varianty uživatelem	Systém umožňuje uživateli vytvořit verzi/variantu jakékoliv instanci objektu JVK, jenž je v instanci objektu Výkaz, který je ve stavu Projektovaný (viz kapitola 2.3 Vazby mezi jednotlivými objekty).	Závazný	1
JVK_5.1	JVK – vytvoření verze/varianty systémem	Systém vytváří novou verzi/variantu JVK pouze pokud je v instanci objektu Výkaz, který je ve stavu Projektovaný. Systém vytváří novou verzi/variantu instance objektu JVK (viz kapitola 2.3 Vazby mezi jednotlivými objekty), pokud byla v systému provedena změna mající vliv pouze na uživatelský tvar JVK.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
JVK_5.2	JVK – změna varianty na verzi	Systém umožňuje uživateli změnit vytvořenou variantu instance objektu JVK na verzi instance objektu JVK (viz OBE_12.0).	Závazný	1
JVK_7.0	JVK – unikátnost atributů	Systém zajišťuje unikátnost atributů kód objektu a název objektu JVK v rámci jedné verze/varianty Výkazu. Nepovolí uživateli založit JVK s kódem objektu nebo názvem objektu, který je již použit pro jinou JVK v rámci jedné verze/varianty Výkazu. Nepovolí uživateli změnit kód objektu nebo název objektu u existující JVK na hodnotu, která je použita pro jinou JVK v rámci jedné verze/varianty Výkazu. Kód objektu lze změnit jen v případě, že existuje pouze první verze a Výkaz je ve stavu Projektovaný.	Závazný	1
JVK_8.0	JVK – ukončení platnosti uživatelem	Systém umožňuje uživateli ukončit platnost instanci objektu JVK (viz OBE_11.0).	Závazný	1
JVK_8.1	JVK – ukončení platnosti systémem v závislosti na zplanění jiné verze/varianty téhož objektu	Systém automaticky ukončí platnost instanci objektu JVK (viz OBE_11.1).	Závazný	1
JVK_8.2	JVK – prodloužení platnosti	Systém prodlužuje platnost instance objektu JVK v rámci prodloužení platnosti nadřízené instance objektu Výkaz (viz VYK_5.2).	Závazný	1
JVK_8.3	JVK – smazání	Systém umožňuje uživateli smazat instanci objektu JVK (viz OBE_10.0)	Závazný	1
JVK_9.0	JVK - operátory	Systém disponuje pro tvorbu JVK operátory uvedenými v kapitole 3.19.3.1 Kontroly vytvořené sémantickým jazykem).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
JVK_9.1	JVK – import algoritmu JVK	Systém umožňuje uživateli importovat algoritmus pro JVK (viz kapitola 5.5.2.3 Věcné kontroly zapsané algoritmem).	Závazný	3
JVK_9.2	JVK – seznam algoritmů pro JVK	Systém umožňuje uživateli zobrazit seznam algoritmů pro JVK, které byly do systému uloženy podle JVK_9.1 ve formě tabulky (grid).	Závazný	3
JVK_9.3	JVK – export algoritmu do textu	Systém umožňuje uživateli exportovat algoritmus JVK do textového souboru.	Závazný	3
JVK_10.0	JVK – chyba v sémantickém tvaru kontroly	Systém informuje uživatele na chybu v sémantickém tvaru JVK. Zjištěnou chybu v sémantickém tvaru JVK barevně vyznačí v zápise sémantického tvaru JVK.	Závazný	1
JVK_10.1	JVK – vytvoření uživatelského tvaru	Systém vytváří uživatelský tvar JVK, jež je zapsána sémantickým jazykem. Systém nevytváří uživatelský tvar JVK, pokud sémantický tvar kontroly obsahuje chyby. Pokud úpravou metapopisu dojde k změně lokace Údaje v rámci Datové oblasti bez dopadu na sémantický tvar JVK, systém vytváří novou verzi/variantu JVK (viz JVK_5.1) a vytváří této JVK nový uživatelský tvar.	Závazný	1
JVK_10.2	JVK – nezadaná odchylka	Systém informuje uživatele na chybějící odchylku JVK v sémantickém tvaru JVK.	Závazný	1
JVK_11.0	JVK – text chybového hlášení	Systém na základě definovaného algoritmu (viz dokument D – Sběr dat kapitola Protokol o dokončení zpracování Vstupní zprávy) vytváří text chybového hlášení, které se odesílá v rámci Výstupní zprávy (viz	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		dokument D – Sběr dat, objekt Výstupní zpráva).		
JVK_12.0	JVK – výměna Parametru v sémantickém tvaru	Systém umožňuje uživateli provést hromadnou výměnu Parametru v sémantickém tvaru vybraných JVK.	Závazný	3
JVK_13.0	JVK – zobrazení seznamu podle Výkazu	Systém umožňuje uživateli zobrazit seznam všech instancí objektu JVK v rámci jedné instance objektu Výkaz ve formě tabulky (grid).	Závazný	1
JVK_13.1	JVK – zobrazení seznamu podle Datové oblasti	Systém umožňuje uživateli zobrazit seznam všech instancí objektu JVK v rámci jedné instance objektu Datová oblast ve formě tabulky (grid).	Závazný	1
JVK_13.2	JVK – zobrazení zápisu Jednovýkazové kontroly	Systém umožňuje uživateli zobrazit zápis JVK zobrazené podle JVK_13.0 nebo JVK_13.1. Zobrazený zápis obsahuje sémantický a uživatelský tvar včetně vyznačení Údajů, vstupujících do JVK, ve struktuře Datových oblastí instance objektu Výkaz.	Závazný	1
JVK_14.0	JVK – prezentace Osobám	Systém prezentuje JVK Osobám v rámci prezentace instance objektu Výkaz (viz VYK_8.1).	Závazný	2
JVK_15.0	JVK – hromadné akce – úpravy obsahu kontrol a kopírování kontrol	Systém umožňuje uživateli hromadné úpravy kontrol (sémantického tvaru), hromadné duplikování/kopírování kontrol z jednoho výkazu do jiných výkazů.	Závazný	1
JVK_16.0	JVK – hromadné akce – úpravy názvů kontrol, odchylek a	Systém umožňuje uživateli hromadné úpravy: - názvů kontrol, - odchylek v kontrolách,	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	úrovně závažnosti kontrol.	- úrovně závažnosti kontrol.		

7.21 Mezivýkazová kontrola (MVK)

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
MVK_21.0	MVK – sémantický jazyk MVK	Systém umožňuje uživateli zapsat MVK prostřednictvím sémantického jazyka pro tvorbu kontrol (viz kapitola 3.19.2.1 Kontroly vytvořené sémantickým jazykem a kapitola 3.19.5.1 Kontroly vytvořené sémantickým jazykem).	Závazný	1
MVK_21.1	MVK – algoritmická	Systém umožňuje uživateli vytvořit algoritmickou MVK (viz kapitola 3.19.5.2 Algoritmické kontroly) výběrem algoritmu pro MVK ze seznamu (viz MVK_29.2) a jeho aplikací na konkrétní Údaje.	Závazný	2
MVK_21.2	MVK – sledování historie	Systém u každé instance objektu MVK sleduje její historii podle kapitoly 3.19.5 Objekt Mezivýkazová kontrola (MVK).	Závazný	1
MVK_21.3	MVK – vytvoření algoritmu dodavatelem	Systém je dodán včetně algoritmů pro MVK uvedené v 8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systém SDAT.	Závazný	2
MVK_22.0	MVK – Skupina MVK	Systém umožňuje uživateli vytvořit Skupinu MVK podle pravidel uvedených v kapitole 3.19.5 Objekt Mezivýkazová kontrola (MVK).	Závazný	1
MVK_23.0	MVK – vytvořena uživatelem	Systém umožňuje uživateli ručně vytvořit MVK, která je popsána základními atributy (viz kapitola 3.19.1 Atributy objektu Kontrola). Systém umožňuje uživateli vytvořit MVK v sémantickém jazyce (viz MVK_21.0) i algoritmickým zápisem (viz MVK_21.1) a to pouze pro	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		instanci objektu Výkaz, jenž se nachází ve stavu Projektovaný (viz kapitola 5.5.2.2 Věcné kontroly vytvářené ručně uživatelem).		
MVK_23.1	MVK – sémantický zápis	Systém umožňuje uživateli vytvořit MVK podle MVK_23.0 přímým zápisem sémantického tvaru kontroly. Před uložením vzorce systém: • provede kontrolu správnosti syntaxe daného zápisu (kompilace vzorce) s ohledem na definici sémantického jazyka. Pokud syntaxe neodpovídá pravidlům sémantického jazyka, pak systém informuje uživatele o všech chybách, které v rámci procesu kompilace daného vzorce nalezl, • provede kontrolu, zda se ve vzorci vyskytují Hodnoty údaje ze všech Výkazů, které jsou definováni jako členové dané Skupiny MVK. Pokud je toto pravidlo porušeno, systém toto vyhodnocuje jako chybu, • provede kontrolu, zda se ve vzorci MVK vyskytují Hodnoty údaje z jiných Výkazů, než jsou definováni jako členové dané MVK. Pokud se ve vzorci objevuje odkaz alespoň na jeden Výkaz, který není členem dané Skupiny MVK, pak toto systém vyhodnocuje jako chybu. Systém umožňuje kdykoli uložit vzorec MVK, a to i tehdy, pokud obsahuje nějakou zjištěnou chybu (kontroly popsány výše). V případě, že systém zjistí, že v okamžiku ukládání vzorec obsahuje identifikovatelnou chybu, upozorní na tuto skutečnost uživatele a nechá ho vybrat, zda si přeje chybu ihned opravit (v takovém případě systém neprovede uložení vzorce MVK, ale umožňuje uživateli vzorec změnit) anebo uložit i přes existenci této chyby. V případě, že uživatele vybere možnost uložit vzorec MVK i když obsahuje chybu, pak systém nastaví atribut MVK validní na hodnotu ne.	Závazný	1
MVK_23.2	MVK – zápis průvodcem	Systém umožňuje uživateli vytvořit MVK pomocí průvodce (viz kapitola 5.5.2.2 Věcné kontroly vytvářené ručně uživatelem). Tento průvodce	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		umožňuje uživateli vybírat porovnávané Hodnoty údaje pomocí uživatelského rozhraní a na základě toho, jak uživatel vybírá a spojuje jednotlivé Hodnoty údaje, na pozadí generuje vzorec v sémantickém jazyce. Systém neumožňuje uživateli vybírat Hodnoty údaje z jiných Výkazů, než z těch, které jsou definované jako členové dané skupiny MVK.		
MVK_23.3	MVK – stanovení odchylky MVK	Systém umožňuje uživateli stanovit odchylku MVK přímým zápisem do sémantického tvaru (viz MVK_23.1) nebo pomocí průvodce (viz MVK_23.2).	Závazný	1
MVK_23.4	MVK – atributy nastavené systémem	Systém nastavuje MVK vytvořeným podle MVK_23.0 následující atributy: • interní identifikátor objektu, • kód objektu, • autor objektu (přihlášený uživatel), • datum vytvoření (aktuální datum), • kdo aktualizoval (přihlášený uživatel), • datum a čas aktualizace (aktuální datum), • platnost_od (platnost_od výkazu, který je vlastníkem MVK), • platnost_do (platnost_do výkazu, který je vlastníkem MVK), • úroveň závažnosti (závažná chyba), • uživatelský tvar kontroly (viz MVK_30.0).	Závazný	1
MVK_23.5	MVK – atributy zadávané uživatelem	Systém umožňuje uživateli vyplnit atributy: • název objektu, • popis objektu, • poznámka,	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		• sémantický tvar vzorce kontroly. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.		
MVK_23.6	MVK – atributy měněné uživatelem	Systém umožňuje uživateli měnit následující atributy MVK (viz kapitola 2.3.6 Rámcové vymezení závislosti objektů) podle MVK_23.4: • kód objektu, • úroveň závažnosti. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	1
MVK_23.7	Vytvoření MVK – Rozpracovaná MVK	Systém umožňuje nastavit jakékoliv MVK atribut dokončená na ne. Takto může být označena jakákoli MVK, která obsahuje validní vzorec, ale z věcného hlediska není kontrola dokončená.	Závazný	1
MVK_24.0	MVK – vytvořené replikací	Systém vytváří MVK pro instance objektů Datová oblast, které vznikly podle DOB_1.1 a DOB_1.2. Takto vytvořené MVK mají stejné atributy (vyjma atribut interní identifikátor) jako MVK, z nichž byly vytvořeny.	Závazný	1
MVK_25.0	MVK – vytvoření verze/varianty uživatelem	Systém umožňuje uživateli vytvořit verzi/variantu jakékoliv instanci objektu MVK, jenž je v instanci objektu Výkaz (označený jako Vlastník MVK), který je ve stavu Projektovaný (viz kapitola 2.3 Vazby mezi jednotlivými objekty). Úprava instance objektu MVK je možná pouze v instanci objektu Výkaz, který je označen jako vlastník MVK. Systém umožňuje editovat instanci objektu MVK, pokud nadřízená instance objektu Skupina MVK není připojena k žádné instanci objektu Plán skupiny MVK pro vykazovací povinnost ani k žádné instanci objektu Plán Skupiny MVK pro výskyt výkazu.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		V případě, že nadřízená instance objektu MVK je připojena alespoň k jedné další instanci výše uvedených souvisejících objektů, pak systém umožňuje editaci instance MVK také provést, ale výsledkem celé akce musí být nová datumová verze objektu Skupina MVK. Pokud je Členem MVK ve Skupině MVK (kam spadá editovaná MVK) Výkaz, který je označen jako Platný, pak v souvislosti s editací dané MVK a jejího zaverzování musí dojít k zaverzování všech dalších souvisejících instancí objektu Výkaz i samotné instance objektu Výkaz.		
MVK_25.1	MVK – vytvoření verze/varianty systémem	Systém vytváří novou verzi/variantu MVK, pouze pokud je v instanci objektu Výkaz označený jako vlastník MVK, který je ve stavu Projektovaný. Systém vytváří novou verzi/variantu instance objektu MVK (viz kapitola 2.3 Vazby mezi jednotlivými objekty), pokud byla v systému provedena změna mající vliv pouze na uživatelský tvar MVK. Úprava instance objektu MVK je možná pouze v instanci objektu Výkaz, který je označen jako vlastník MVK.	Závazný	1
MVK_25.2	MVK – změna varianty na verzi	Systém umožňuje uživateli změnit vytvořenou variantu instance objektu MVK na verzi instance objektu MVK (viz OBE_12.0).	Závazný	1
MVK_27.0	MVK – unikátnost atributů	Systém zajišťuje unikátnost atributů kód objektu a název instance objektu MVK v rámci jedné Skupiny MVK. Nepovolí uživateli založit MVK s kódem objektu nebo názvem objektu, který je již použit pro jinou MVK v rámci jedné Skupiny MVK. Nepovolí uživateli změnit kód objektu nebo název objektu u existující MVK na hodnotu, která je použita pro jinou MVK v rámci jedné Skupiny MVK. Kód objektu lze změnit jen v případě, že existuje pouze první verze, a Výkaz je ve stavu Projektovaný.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
MVK_28.0	MVK – ukončení platnosti uživatelem v závislosti na zplatnění jiné verze/varianty téhož objektu	Systém umožňuje uživateli ukončit platnost instanci objektu MVK (viz OBE_11.0). Ukončení platnosti MVK má tyto následky: • pokud nadřízená Skupina MVK je již připojena k nějakému Výskytu výkazu (zkoumají se pouze Výskyty výkazu, které jsou ve stavu 10 - Připravený), systém informuje uživatele, ke kterým Výskytům výkazu je daná Skupina MVK přiřazena a umožňuje uživateli rozhodnout, pro které Výskyty výkazu se má daná MVK zneplatnit také (pokud toto uživatel explicitně neurčí, má se za to, že bude Skupina MVK zneplatněna u všech Výskytů výkazu, kde se vyskytuje), • pokud nadřízená instance objektu Skupina MVK je již připojena k nějaké Vykazovací povinnosti, pak systém zajistí, že při dalším generování Výskytu výkazu z této Vykazovací povinnosti již nebude daná MVK přiřazena k nově vzniklým Výskytům výkazu. Pokud je Členem MVK ve Skupině MVK (kam spadá editovaná MVK) Výkaz, který je označen jako Platný, pak v souvislosti se zneplatněním dané MVK a jejího zaverzování, musí dojít k zaverzování všech dalších souvisejících instancí objektu Výkaz i samotné instance objektu Výkaz.	Závazný	1
MVK_28.1	MVK – ukončení platnosti systémem	Systém automaticky ukončí platnost instanci objektu MVK (viz OBE_11.1).	Závazný	1
MVK_28.2	MVK – prodloužení platnosti	Systém prodlužuje platnost instance objektu MVK v rámci prodloužení platnosti nadřízené instance objektu Výkaz (viz VYK_5.2).	Závazný	1
MVK_28.3	MVK – smazání	Systém umožňuje uživateli smazat instanci objektu MVK (viz OBE_10.0).	Závazný	1
MVK_29.0	MVK - operátory	Systém disponuje pro tvorbu MVK operátory uvedenými v kapitole	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		3.19.5.1 Kontroly vytvořené sémantickým jazykem.		
MVK_29.1	MVK – import algoritmu MVK	Systém umožňuje uživateli importovat algoritmus pro MVK (viz kapitola 5.5.2.3 Věcné kontroly zapsané algoritmem).	Závazný	3
MVK_29.2	MVK – seznam algoritmů MVK	Systém umožňuje uživateli zobrazit seznam algoritmů MVK, které byly do systému uloženy podle MVK_29.1 ve formě tabulky (grid).	Závazný	3
MVK_29.3	MVK – export algoritmu do textu	Systém umožňuje uživateli exportovat algoritmus MVK do textového souboru.	Závazný	3
MVK_30.0	MVK – vytvoření uživatelského tvaru	Systém vytváří uživatelský tvar MVK, jenž je zapsaná sémantickým jazykem. Systém nevytváří uživatelský tvar MVK, pokud sémantický tvar kontroly obsahuje chyby. Pokud úpravou metapopisu dojde k změně lokace Údaje v rámci Datové oblasti bez dopadu na sémantický tvar MVK, systém vytváří novou verzi/variantu MVK (viz MVK_25.1) a vytváří této MVK nový uživatelský tvar.	Závazný	1
MVK_30.1	MVK – chyba v sémantickém tvaru kontroly	Systém informuje uživatele na chybu v sémantickém tvaru MVK. Zjištěnou chybu v sémantickém tvaru MVK barevně vyznačí v zápise sémantického tvaru MVK.	Závazný	1
MVK_30.2	MVK – nezadaná odchylka	Systém informuje uživatele na chybějící odchylku MVK v sémantickém tvaru MVK.	Závazný	1
MVK_31.0	MVK – text chybového hlášení	Systém na základě definovaného algoritmu (viz dokument D – Sběr dat kapitola Protokol o dokončení zpracování Vstupní zprávy) vytváří text chybového hlášení, které se odesílá v rámci Výstupní zprávy (viz	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		dokument D – Sběr dat, kapitola 2.5 Objekt Výstupní zpráva)		
MVK_32.0	MVK – výměna parametru v sémantickém tvaru	Systém umožňuje uživateli provést hromadnou výměnu Parametru v sémantickém tvaru vybraných MVK.	Závazný	3
MVK_33.0	MVK – zobrazení seznamu podle Výkazu	Systém umožňuje uživateli zobrazit seznam všech instancí objektu MVK v rámci jedné instance objektu Výkaz ve formě tabulky (grid) Jedna instance objektu MVK je součástí seznamu MVK všech instancí objektu Výkaz, které jsou účastníky MVK.	Závazný	1
MVK_33.1	MVK – zobrazení seznamu podle Datové oblasti	Systém umožňuje uživateli zobrazit seznam všech instancí objektu MVK v rámci jedné instance objektu Datová oblast ve formě tabulky (grid).	Závazný	1
MVK_33.2	MVK – zobrazení zápisu Mezivýkazové kontroly	Systém umožňuje uživateli zobrazit zápis MVK zobrazené podle MVK_33.0 nebo MVK_33.1. Zobrazený zápis obsahuje sémantický a uživatelský tvar včetně vyznačení Údajů, vstupujících do MVK, ve struktuře Datových oblastí instancí objektu Výkaz, které jsou účastníky MVK.	Závazný	1
MVK_34.0	MVK – prezentace Osobám	Systém prezentuje MVK Osobám v rámci prezentace instance objektu Výkaz (viz VYK_8.1).	Závazný	2
MVK_35.0	MVK – hromadné akce – úpravy obsahu kontrol a kopírování kontrol	Systém umožňuje uživateli hromadné úpravy kontrol (sémantického tvaru), hromadné duplikování/kopírování kontrol z jednoho výkazu do jiných výkazů.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
MVK_36.0	MVK – hromadné akce – úpravy názvů kontrol, odchylek a úrovně závažnosti kontrol.	Systém umožňuje uživateli hromadné úpravy: - názvů kontrol, - odchylek v kontrolách, - úrovně závažnosti kontrol.	Závazný	1

7.22 Kontrola časové řady (KČŘ)

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
KČŘ_1.0	KČŘ – algoritmická	Systém umožňuje uživateli vytvořit KČŘ (viz kapitola 3.19.4 Objekt Kontrola časové řady (KČŘ)) výběrem algoritmu KČŘ ze seznamu (viz KČŘ_6.1) a jeho aplikací na konkrétní Údaj nebo Údaje.	Závazný	2
KČŘ_1.1	KČŘ – sledování historie	Systém u každé instance objektu KČŘ sleduje její historii podle kapitoly 3.19.4 Objekt Kontrola časové řady (KČŘ).	Závazný	2
KČŘ_1.2	KČŘ – vytvoření algoritmu dodavatelem	Systém je dodán včetně algoritmu pro KČŘ uvedeného v příloze 8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systém SDAT.	Závazný	2
KČŘ_2.0	KČŘ – atributy nastavené systémem	Systém nastavuje KČŘ vytvořeným podle KČŘ_1.0 následující atributy: • interní identifikátor objektu, • kód objektu, • název objektu, • autor objektu (přihlášený uživatel), • datum vytvoření (aktuální datum),	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		• kdo aktualizoval (přihlášený uživatel), • datum a čas aktualizace (aktuální datum), • platnost_od (platnost_od výkazu), • platnost_do (platnost_do výkazu), • úroveň závažnosti (závažná chyba, pro kontrolu lineární regrese chyba k potvrzení).		
KČŘ_2.1	KČŘ – atributy zadávané uživatelem	Systém umožňuje uživateli vyplnit atributy: • popis objektu, • poznámka. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	2
KČŘ_2.2	KČŘ – atributy (zadané systémem) měněné uživatelem	Systém umožňuje uživateli měnit následující atributy KČŘ (viz kapitola 2.3.6 Rámcové vymezení závislosti objektů) podle KČŘ_2.0: • kód objektu, • název objektu, • úroveň závažnosti. Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole 2.3 Vazby mezi jednotlivými objekty.	Závazný	2
KČŘ_3.0	KČŘ – vytvoření verze/varianty uživatelem	Systém umožňuje uživateli vytvořit verzi/variantu jakékoliv instanci objektu KČŘ, jenž je v instanci objektu Výkaz, který je ve stavu Projektovaný (viz kapitola 2.3 Vazby mezi jednotlivými objekty).	Závazný	2
KČŘ_3.1	KČŘ – změna varianty na verzi	Systém umožňuje uživateli změnit vytvořenou variantu instance objektu KČŘ na verzi instance objektu KČŘ (viz OBE_12.0).	Závazný	2
KČŘ_4.0	KČŘ – unikátnost	Systém zajišťuje unikátnost atributů kód objektu a název instance objektu	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	atributů	KČŘ v rámci jedné verze/varianty Výkazu. Nepovolí uživateli založit KČŘ s kódem objektu nebo názvem objektu, který je již použit pro jinou KČŘ v rámci jedné verze/varianty Výkazu. Nepovolí uživateli změnit kód objektu nebo název objektu u existující KČŘ na hodnotu, která je použita pro jinou KČŘ v rámci jedné verze/varianty Výkazu. Kód objektu lze změnit jen v případě, že existuje pouze první verze, a Výkaz je ve stavu Projektovaný.		
KČŘ_5.0	KČŘ – ukončení platnosti uživatelem	Systém umožňuje uživateli ukončit platnost instanci objektu KČŘ (viz OBE_11.0).	Závazný	2
KČŘ_5.1	KČŘ – ukončení platnosti systémem v závislosti na zplanění jiné verze/varianty téhož objektu	Systém automaticky ukončí platnost instanci objektu KČŘ (viz OBE_11.1).	Závazný	2
KČŘ_5.2	KČŘ – prodloužení platnosti	Systém prodlužuje platnost instance objektu KČŘ v rámci prodloužení platnosti nadřízené instance objektu Výkaz (viz VYK_5.2).	Závazný	2
KČŘ_5.3	KČŘ – smazání	Systém umožňuje uživateli smazat instanci objektu KČŘ (viz OBE_10.0).	Závazný	2
KČŘ_6.0	KČŘ – import algoritmu KČŘ	Systém umožňuje uživateli importovat algoritmus pro KČŘ v souladu s kapitolou 5.5.2.3 Věcné kontroly zapsané algoritmem.	Závazný	3
KČŘ_6.1	KČŘ – seznam algoritmů KČŘ	Systém umožňuje uživateli zobrazit seznam algoritmů KČŘ, které byly do systému uloženy podle KČŘ_6.0 ve formě tabulky (grid).	Závazný	2
KČŘ_6.2	KČŘ – export	Systém umožňuje uživateli exportovat algoritmus KČŘ do textového	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	algoritmu do textu	souboru.		
KČŘ_7.0	KČŘ – text chybového hlášení	Systém na základě definovaného algoritmu (viz dokument D – Sběr dat, kapitola Protokol o dokončení zpracování Vstupní zprávy) vytváří text chybového hlášení, které se odesílá v rámci Výstupní zprávy.	Závazný	2
KČŘ_8.0	KČŘ – zobrazení seznamu podle Výkazu	Systém umožňuje uživateli zobrazit seznam všech instancí objektu KČŘ v rámci jedné instance objektu Výkaz ve formě tabulky (grid).	Závazný	2
KČŘ_8.1	KČŘ – zobrazení seznamu podle Datové oblasti	Systém umožňuje uživateli zobrazit seznam všech instancí objektu KČŘ v rámci jedné instance objektu Datová oblast ve formě tabulky (grid).	Závazný	2
KČŘ_9.0	KČŘ – prezentace Osobám	Systém prezentuje KČŘ Osobám v rámci prezentace instance objektu Výkaz (viz VYK_8.1).	Závazný	2
KČŘ_10.0	KČŘ – diagnostická funkce	Systém umožňuje uživateli spustit diagnostickou funkci, která na základě definovaného algoritmu (viz kapitola 5.5.2.3 Věcné kontroly zapsané algoritmem) vypočte optimální počet kontrol lineární regrese v rámci jedné instance objektu Výkaz. Systém na základě této diagnostiky označí Údaje, pro které je optimální vytvořit kontrolu lineární regrese.	Závazný	2
KČŘ_11.0	KČŘ – Mezisubjektová kontrola	Systém umožňuje uživateli vytvořit speciální typ KČŘ, tzv. Mezisubjektovou kontrolu (viz kapitola 3.19.4.1 Objekt Mezisubjektová kontrola (MSK) jako specifický typ KČŘ).	Závazný	2

7.23 Knihovna

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
KNH_1.0	Zobrazení objektů Knihovny	Systém umožňuje uživateli zobrazit náhled na instance objektů Knihovny (viz kapitola 4 Knihovna).	Závazný	1
KNH_2.0	Navigace v Knihovně	Systém umožňuje uživateli navigaci prostřednictvím volby časového řezu jako základní: 1) „poslední“ – zobrazují se pouze poslední instance (nejvyšší číslo (objektů): a) pokud došlo k ukončení poslední verze/varianty objektu, zobrazuje se jeho poslední verze/varianta, i když v aktuálním čase je neplatná, b) pokud došlo k ukončení poslední verze/varianty objektu a její platnost_od je menší než aktuální datum, nezobrazuje se, 2) „časový řez“ (typováním, výběrem z nabídky z listu hodnot platnost_od) – zobrazení instancí objektů, jejichž interval platnost_od a platnost_do zahrnuje zvolený časový řez: a) zobrazí se všechny instance (vzhledem k ukončování platnosti při zplatnění od jednoho objektu může být více instancí), b) ze zobrazení se vyloučí ty, jejichž platnost_do bude při zplatnění zkrácena na hodnotu menší nebo rovno než je konkrétní časový řez (od každého objektu bude max. jedna instance).	Závazný	1
KNH_2.1	Uživatelské nastavení navigace	Systém umožňuje uživateli nastavit „svůj“ default navigace v Knihovně.	Závazný	2
KNH_3.0	Historie objektů Knihovny	Systém umožňuje uživateli zobrazit k objektu jeho historii, tj. jednotlivé instance objektu, jejich platnost_od a platnost_do, autor objektu, datum vytvoření, kdo aktualizoval, datum a čas aktualizace).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Z historie objektu systém umožňuje uživateli zobrazit detail instance, kterou vybere.		
KNH_4.0	Historie atributů objektů Knihovny	Systém umožňuje uživateli zobrazit historii jednotlivých atributů a vazeb.	Závazný	1
KNH_5.0	Použití objektů Knihovny	Systém umožňuje uživateli zobrazit použití jednotlivých instancí objektů v nadřízených objektech: • v časovém řezu objektu, • do budoucna, podle kapitoly 2.4.1 Základní vlastnosti objektů. Systém zobrazuje použití objektu v tabulce (grid). Uživatel může provádět filtry a zobrazit detail vybraného objektu.	Závazný	1
KNH_6.0	Nepoužité objekty Knihovny	Systém umožňuje uživateli zobrazit instance objektů, které nemají použití v nadřízených objektech: • v časovém řezu objektu, • do budoucna. Systém zobrazuje nepoužité objekty v tabulce (grid). Uživatel může provádět filtry a zobrazit detail vybraného objektu.	Závazný	1

7.24 Navazování časových řad

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
CAR_1.0	Navazování	Systém umožňuje uživateli navázat do časové řady Údaje dle kapitoly 6.2	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	časové řady Údaje	Proces Navazování časových řad Údajů.		
CAR_2.0	Zajištění časových řad při nezměněném popisu Údaje	Systém automaticky zajišťuje časové řady Údajů, které v jednotlivých instancích Výkazu nezměnily popis.	Závazný	3
CAR_3.0	Kontrola navázání časové řady	Systém umožňuje uživateli spustit kontrolu správnosti navázání časové řady Údajů (viz kapitola 6.2.2 Průběh procesu).	Závazný	3
CAR_4.0	Zobrazení přerušených časových řad Údajů	Systém umožňuje uživateli zobrazit přerušené časové řady Údajů v grafickém vzoru Výkazu.	Závazný	3
CAR_4.1	Zobrazení hloubek časových řad Údajů	Systém umožňuje uživateli zobrazit začátek a konec časových řad Údajů v grafickém vzoru Výkazu a při výběru Hodnot údajů.	Závazný	3
CAR_4.2	Zobrazení Časové řady Údaje	Systém umožňuje uživateli zobrazit řetězec časové řady Údaje.	Závazný	3
CAR_5.0	Rozvázání časové řady	Systém umožňuje uživateli rozpojit jednou navázané časové řady Údajů.	Závazný	3

7.25 Grafické zobrazení struktury

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
GZS_1.0	Grafické zobrazení struktury – vizualizace DO dle metapopisu	Systém umožňuje uživateli vizualizovat Datové oblasti z nadeklarovaného metapopisu ve formulářovém prostředí. Grafické zobrazení struktury umožňuje kromě vizualizace DO z deklarovaného metapopisu zobrazit také náhled struktury, která abstrahuje od nastavení chtěných či nechtěných produktů kartézského součinu na jednotlivých osách a nastavení vlastního pořadí jednotlivých produktů na jednotlivých osách.	Závazný	1
GZS_2.0	Grafické zobrazení struktury – úprav nemající vliv na metapopis	Systém umožňuje uživateli udělat takové grafické úpravy, které nemají vliv na metapopis. Uživateli je umožněno přesouvat řádky a sloupce a nastavovat jim jiné pořadí. Tato akce okamžitě změny promítá do příslušného formuláře, kde se tato pořadí nastavují deklarativně. Dále je uživateli umožněno mezi řádky či sloupce vložit textovou informaci, která není tvořena metapopisem, nemá vliv na stávající Údaje Datové oblasti či další generování Údajů.	Závazný	1
GZS_3.0	Grafické zobrazení struktury – nastavení atributů Údaje	Systém umožňuje uživateli nastavit hodnoty atributům na úrovni jednotlivých Údajů ve struktuře Datové oblasti. Uživateli je umožněno nastavit atributy Údaje, a to: ○ stupeň citlivosti údaje, ○ vykazovat, ○ zobrazovaný údaj.	Závazný	1
GZS_4.0	Grafické zobrazení struktury – úprava názvů ve struktuře	Systém umožňuje uživateli volitelnou úpravu názvů (ukazatelů, parametrů a položek parametrů) objektů, které se v Datové oblasti vizualizují z nadeklarovaného metapopisu. Úprava názvů je platná pouze pro danou Datovou oblast, touto úpravou se nemění názvy objektů v Knihovně. Volitelná úprava názvů objektů v grafickém zobrazení struktury v podstatě upravuje „text ve struktuře“ vizualizované Datové	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		oblasti Upravené názvy jsou pak dále prezentovány Vykazujícím osobám.		
GZS_5.0	Grafické zobrazení struktury – zamezení automatických kontrol pro Údaje	Systém umožňuje uživateli označení Údaje a hierarchie, pro které se nevygenerují automatické kontroly. Údaje a hierarchie může uživatel označit jednotlivě i hromadně.	Závazný	1

7.26 Načítání artefaktů z externího metapopisu

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
S2X_1.0	Založení objektů Knihovny importem z externího metapopisu	Systém umožňuje automatizované zakládání objektů Knihovny v rámci procesu načítání externího metapopisu dle kapitoly 6.1 Proces Přebíraní metapopisu z externích zdrojů.	Závazný	2
S2X_2.0	Založení Výkazů importem z externího metapopisu	Systém umožňuje automatizované zakládání Výkazů vč. jejich struktury, v rámci procesu načítání externího metapopisu dle kapitoly 6.1 Proces Přebíraní metapopisu z externích zdrojů. Struktura výkazů je odvozena z vrstvy Table Linkbase vytvořené dle specifikace http://specifications.xbrl.org/work-product-index-table-linkbase-table-linkbase-1.0.html, která je součástí XBRL taxonomií.	Závazný	2
S2X_3.0	Import kontrol – XBRL	Systém umožňuje automatizované zakládání kontrol v rámci procesu načítání externího metapopisu dle kapitoly 6.1 Proces Přebíraní	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		metapopisu z externích zdrojů.		
S2X_4.0	Import kontrol – XLS, XLSX	Systém umožňuje dodatečné inkrementální importy konktrol na základě doprovodných souborů XLS, XLSX, které nejsou přímou součástí XBRL taxonomií.	Závazný	2

7.27 Speciální kontroly (MKT)

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
SPK_1.0	Implementace speciálních kontrol (MKT)	V systému jsou implementované všechny speciální kontroly dle kapitoly 8.1.2 Kontroly v systému kapitálových trhů (MKT).	Závazný	2

8 Přílohy

8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systém SDAT

8.1.1 Kontroly v časové řadě (KČŘ)

Kontrola odchýlení hodnoty údaje od lineární regresní přímky používá funkci, která zjišťuje, zda Hodnota údaje ze zpracovávaného Vydání výskytu výkazu leží v intervalu spolehlivosti hodnoty predikované na základě minulých hodnot. Principem metody je proložení přímky posledními 3 až 5 hodnotami a odvození přípustné odchylky na základě metody lineární regresní analýzy. Systém umožňuje stanovit Údaje, u jejichž hodnot bude při zpracování prováděna kontrola vůči časovým řadám. Kontrola vůči časovým řadám se definuje u statických údajů. Systém stanovuje optimální počet kontrol (tj. počet Údajů, u kterých se bude provádět) v časové řadě, podrobněji viz kapitola 5.5 Proces tvorba kontrol výkazu. Prostřednictvím technické správy aplikace je umožněna parametrizace vybrané statistické metody.

Kontrola jedinečnosti hodnoty Údaje zjišťuje, zda se Hodnota údaje ze zpracovávaného Vydání výskytu výkazu neshoduje s některou z Hodnot údaje kteréhokoli z předchozích Vydání výskytu výkazu, jež obsahují tento Údaj. Je nutné také vzít v úvahu, že ten samý Údaj mohl být v minulosti součástí jiného Výkazu. Pokud je shoda nalezena, kontrola ohlásí chybu s úrovní závažnosti závažná chyba.

Mezisubjektové kontroly (MSK) pro systém MKT (Monitoring kapitálového trhu)

Tento typ kontrol je použit mezi Výkazy zaslanými organizátorem regulovaného trhu, tzn. Burzou cenných papírů Praha (dále jen BCPP), a výkazy zaslanými obchodníky s cennými papíry (dále jen OCP) podnikající na kapitálovém trhu dle zákona č. 256/2004 Sb., o podnikání na kapitálovém trhu.

Kontroly mají za úkol identifikovat nekonzistenci vykázaných transakcí, které byly/měly být uskutečněny na regulovaném trh BCPP a měly být vykázány jak organizátorem trhu, tak i zúčastněným OCP, a to na úrovni identifikátorů transakcí jako je např. číslo obchodu, číslo objednávky, číslo pokynu atd.

Kontroly musí probíhat křížově, např.:

- obchody vykázané na straně organizátora trhu musí mít související záznam na straně OCP,
- obchody vykázané na straně OCP musí mít související záznam na straně organizátora trhu.

Do tohoto typu kontrol vstupují Výkazy, které jsou zasílány regulátorem trhu na denní bázi, proti Výkazům, které jsou zasílány na měsíční bázi ze strany OCP, tedy jedno Vydání výskytu výkazu (měsíční Výkaz dodaný OCP) proti N Vydáním výskytu výkazu (sada denních Výkazů dodaných organizátorem trhu).

V případě, že dojde k nekonzistenci (např. na straně OCP buď neexistuje ID obchodu, které existuje na straně organizátora trhu, anebo existuje takové ID obchodu na straně OCP, které

neexistuje na straně organizátora trhu), kontrola ohlásí chybu s úrovní závažnosti chyba k potvrzení, a to pouze u výkazů od OCP.

Kontroly jsou spouštěny vždy při příjmu Vydání výskytu výkazu ze strany OCP. Při tomto typu kontroly se vychází z předpokladu, že data poskytovaná organizátorem trhu jsou „referenční“, tedy bezchybná. Veškeré nalezené chyby jsou tak vztaženy a komunikovány směrem k OCP.

V rámci kontroly je nutno zajistit, že kontrola může probíhat i mimo periodu jednoho měsíce.

8.1.2 Kontroly v systému kapitálových trhů (MKT)

Funkční oblast Kapitálové trhy v současnosti jako jediná využívá kromě tradiční interpretace kontrol interním interpretem i možnost interpretovat kontroly externě přes Standardní Programátorské rozhraní (SPR) MTS. SPR je pro tento účel vybaveno objekty, které publikují kontrolovaný obsah mimo MTS a přijímají informace o výsledcích provedených kontrol. Interní interpret je v metodice označen zkratkou JVK, externí pak MKT. V metodice MKT20150101 je celkem 188 kroků JVK. Z toho jich 76 (40 %) zajišťuje interpret JVK a 112 (60 %) interpret MKT. Rozpis po výkazech je v příloze č. 1. Dále metodika MKT20150101 obsahuje 18 kroků MVK. Z toho 7 jich vlastní MOKAS41 (Pokyny) a 11 MOKAS42 (Obchody). Závislými výkazy jsou MOKAS41, 43 a 44. Všechny tyto MVK jsou interpretovány externě. Třídy kontrol jsou popsány v kapitole 8.1.2.1 Třídy externích kontrol v systému kapitálových trhů (MKT), algoritmy konkrétních jednotlivých kontrol pak v kapitole 8.1.2.2 Popis externích kontrol v systému kapitálových trhů (MKT).

8.1.2.1 Třídy externích kontrol v systému kapitálových trhů (MKT)

Externí MVK využívají organizační logiku standardních MVK (plánování, protokolace, atd.), pracují ovšem převážně s daty, která nejsou v dosahu současného systému MtS. Tato data mají také zpravidla objemy, které přesahují současné možnosti interního zpracování v současném systému MtS. Kritériem pro zařazení konkrétní kontroly do určité třídy kontrol byl typ jejího algoritmu. Následuje výpis všech tříd kontrol s popisem a uvedením výkazů, ve kterých jsou ve stavájícím systému MKT implementovány:

Třída	Popis
A	Kontrola globální unikátnosti skupiny hodnot. Kontrola pracuje s daty všech výskytů (MOKAS40, MOKAS50).
B	Kontrola unikátnosti skupiny hodnot v rozsahu výskytu nebo vydání. Určený rozsah je uveden v názvu kontroly (JISIFE51, JISIFE52, JISIFE53, MOKAS40, MOKAS41, MOKAS42, MOKAS43, MOKAS44, MOKAS60, MOKAS61, MOKAS62, MOKAS63, MOKAS65, MOKAS70, MOKAS80).
C	Kontrola, zda hodnota odpovídá typu, který subjekt určil z daných možností. Typický název kontroly: "Hodnota údaje X není v souladu s vykázaným Typem X" (JISIFE51, JISIFE52, JISIFE53, MOKAS40, MOKAS43, MOKAS44, MOKAS50, MOKAS60, MOKAS61, MOKAS62, MOKAS63, MOKAS65, MOKAS70, MOKAS80).
D	Podmíněně prováděné kontrola (JISIFE51, MOKAS42, MOKAS60).
E	Kontrola, zda hodnota odpovídá typu, který předepisuje metodika. Typický název kontroly: "Hodnota údaje X není (typu) Y" (MOKAS40, MOKAS43,

Třída	Popis
	MOKAS44, MOKAS50, MOKAS60, MOKAS62, MOKAS80).
F	Kontrola, zda hodnota existuje v externím číselníku (MOKAS40, MOKAS42, MOKAS50, MOKAS60).
G	Kontrola hodnoty vůči strojovému času (MOKAS40, MOKAS41, MOKAS42, MOKAS44, MOKAS50, MOKAS80).
H	Kontrola mezi řádky typu rodič a potomek (MOKAS41, MOKAS42).
I	Kontrola existence odkazu do globálního seznamu hodnot. Kontrola pracuje s daty všech výskytů (JISIFE51, JISIFE52, JISIFE53, MOKAS40, MOKAS41, MOKAS42, MOKAS43, MOKAS44, MOKAS50).
J	Kontrola hodnoty vůči jednomu výskytu. Je to vlastně klasická MVK prováděná externě (MOKAS41, MOKAS42).
K	Kontrola hodnoty vůči více (alternativním/externím) číselníkům (MOKAS41, MOKAS42).
L	Kontrola hodnoty současně vůči více výskytům výkazu (MOKAS42).
M	Kontrola hodnoty vůči hodnotě vypočtené pomocí referenčního výskytu (MOKAS42).

Tabulka 6 - Třídy kontrol v MKT

Každá z uvedených kontrol může být prováděna podmíněně, kromě své vlastní třídy pak náleží také do třídy "D". Samotná třída "D" řeší zpravidla nepřítomnost příslušného podmínkového aparátu interně v MTS.

8.1.2.2 Popis externích kontrol v systému kapitálových trhů (MKT)

Přehled kontrol metodiky MKT20150101 v systému kapitálových trhů (MKT) včetně jejich popisu obsahuje Tabulka 7 - Popis kontrol v MKT. Přehled je setříděný podle klasifikace tříd, kterou uvádí Tabulka 6 - Třídy kontrol v MKT.

Třída	ID výkazu	ID kontroly	Název kontroly
A	MOKAS40	200	Kombinace hodnot údajů Referenční číslo obchodu (EKT0003), Autor čísla obchodu (EKT0004) a Ukazatel nákupu/prodeje (OKT0001) musí být unikátní. Kontrolují se jen nezrušené řádky.
A	MOKAS40	30	Kombinace hodnot údajů Referenční číslo hlášení (R0002) a Autor čísla hlášení (R0003) musí být unikátní.
A	MOKAS50	200	Kombinace hodnot údajů Referenční číslo obchodu (EKT0003), Autor čísla obchodu (EKT0004) a Ukazatel nákupu/prodeje (OKT0001) musí být unikátní. Kontrolují se jen nezrušené řádky.
A	MOKAS50	30	Hodnota údaje Referenční číslo hlášení (R0002) musí být unikátní.
B	JISIFE51	20	Kombinace hodnot údajů Kód cenného papíru (P0313), Struktura sledovaných měn (P0019), Identifikace klienta

Třída	ID výkazu	ID kontroly	Název kontroly
			(P0307) a Typ vztahu na klienta (P1505) musí být unikátní v rámci výskytu.
B	JISIFE52	60	Kombinace hodnot údajů Identifikace investičního nástroje (EKT0107), Identifikace klienta (P0307) a Typ vztahu na klienta (P1505) musí být unikátní v rámci výskytu.
B	JISIFE53	50	Kombinace hodnot údajů Identifikace klienta (P0307) a Typ vztahu na klienta (P1505) musí být unikátní v rámci výskytu.
B	MOKAS41	50	Hodnota údaje Číslo pokynu (EKT0049) musí být unikátní v rámci výskytu. Kontrolují se jen řádky s nevyplněnou hodnotou údaje Důvod zrušení (EKT0065).
B	MOKAS42	70	Kombinace hodnot údajů Referenční číslo obchodu (EKT0003), Autor čísla obchodu (EKT0004) a Ukazatel nákupu/prodeje (OKT0001) musí být unikátní v rámci výskytu. Kontrolují se jen řádky s nevyplněnou hodnotou údaje Datum zrušení obchodu (DKT0030).
B	MOKAS43	20	Hodnota údaje Identifikace investičního nástroje (EKT0107) musí být unikátní v rámci výskytu.
B	MOKAS44	30	Hodnota údaje Identifikace osoby (R0010) musí být unikátní v rámci výskytu.
B	MOKAS60	10	Hodnota údaje Jednoznačná identifikace záznamu (R0007) musí být unikátní v rámci výskytu.
B	MOKAS60	40	Hodnota údaje Identifikace investičního nástroje (EKT0107) musí být unikátní v rámci výskytu.
B	MOKAS61	10	Hodnota údaje Jednoznačná identifikace záznamu (R0007) musí být unikátní v rámci výskytu.
B	MOKAS62	10	Hodnota údaje Jednoznačná identifikace záznamu (R0007) musí být unikátní v rámci výskytu.
B	MOKAS63	10	Hodnota údaje Jednoznačná identifikace záznamu (R0007) musí být unikátní v rámci výskytu.
B	MOKAS65	10	Hodnota údaje Jednoznačná identifikace záznamu (R0007) musí být unikátní v rámci výskytu.
B	MOKAS70	10	Hodnota údaje Jednoznačná identifikace záznamu (R0007) musí být unikátní v rámci výskytu.
B	MOKAS80	10	Hodnota údaje Jednoznačná identifikace záznamu (R0007) musí být unikátní v rámci výskytu.
C	JISIFE51	10	Hodnota údaje Kód cenného papíru (P0313) není v souladu s vykázaným Typem kódu cenného papíru (P1094).
C	JISIFE51	190	Hodnota údaje Identifikace klienta (P0307) není v souladu s vykázaným Typem identifikace osoby = BIC|IČ|IID (P0523= BIC|IC|IID).
C	JISIFE51	200	Hodnota údaje Identifikace klienta (P0307) není v souladu s vykázaným Typem identifikace osoby =

Třída	ID výkazu	ID kontroly	Název kontroly
			NID|RČ (P0523= NID|RC).
C	JISIFE52	30	Hodnota údaje Identifikace klienta (P0307) není v souladu s vykázaným Typem identifikace osoby = BIC|IČ|IID (P0523= BIC|IC|IID).
C	JISIFE52	40	Hodnota údaje Identifikace klienta (P0307) není v souladu s vykázaným Typem identifikace osoby = NID|RČ (P0523= NID|RC).
C	JISIFE52	50	Hodnota údaje Identifikace investičního nástroje (EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	JISIFE53	30	Hodnota údaje Identifikace klienta (P0307) není v souladu s vykázaným Typem identifikace osoby = BIC|IČ|IID (P0523= BIC|IC|IID).
C	JISIFE53	40	Hodnota údaje Identifikace klienta (P0307) není v souladu s vykázaným Typem identifikace osoby = NID|RČ (P0523= NID|RC).
C	MOKAS40	100	Hodnota údaje Protistrana (EKT0001) není v souladu s vykázaným Typem identifikace protistrany = IČO|NID|RČ (EKT0181 = IC|NID|RC).
C	MOKAS40	140	Hodnota údaje Identifikace místa (EKT0002) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	MOKAS40	160	Hodnota údaje Identifikace zákazníka (EKT0005) není v souladu s vykázaným Typem identifikace zákazníka = BIC|Interní identifikace (EKT0182 = BIC|IID).
C	MOKAS40	170	Hodnota údaje Identifikace zákazníka (EKT0005) není v souladu s vykázaným Typem identifikace zákazníka = IČO|NID|RČ (EKT0182 = IC|NID|RC).
C	MOKAS40	60	Hodnota údaje Identifikace investičního nástroje (EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	MOKAS40	90	Hodnota údaje Protistrana (EKT0001) není v souladu s vykázaným Typem identifikace protistrany = BIC|Interní identifikace|MIC (EKT0181 = BIC|IID|MIC).
C	MOKAS43	10	Hodnota údaje Identifikace investičního nástroje (EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	MOKAS43	30	Hodnota údaje Identifikace podkladového nástroje (EKT0110) není v souladu s vykázaným Typem identifikace podkladového nástroje (EKT0111).
C	MOKAS44	10	Hodnota údaje Identifikace osoby (R0010) není v souladu s vykázaným Typem identifikace osoby = BIC (P0523 = BIC). Kontrolují se jen řádky s nevyplněnou hodnotou údaje Autor identifikace osoby (EKT0061).
C	MOKAS44	20	Hodnota údaje Identifikace osoby (R0010) není v

Třída	ID výkazu	ID kontroly	Název kontroly
			souladu s vykázaným Typem identifikace osoby = IČO|NID|RČ (P0523 = IC|NID|RC). Kontrolují se jen řádky s nevyplněnou hodnotou údaje Autor identifikace osoby (EKT0061).
C	MOKAS50	100	Hodnota údaje Protistrana (EKT0001) není v souladu s vykázaným Typem identifikace protistrany = BIC|Interní identifikace|MIC (EKT0181 = BIC|IID|MIC).
C	MOKAS50	110	Hodnota údaje Protistrana (EKT0001) není v souladu s vykázaným Typem identifikace protistrany = IČO|NID|RČ (EKT0181 = IC|NID|RC).
C	MOKAS50	150	Hodnota údaje Identifikace místa (EKT0002) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108). Kontrolují se jen řádky s hodnotou údaje Hlášení OCP (OKT0004) = Y.
C	MOKAS50	160	Hodnota údaje Identifikace zákazníka (EKT0005) není v souladu s vykázaným Typem identifikace zákazníka = BIC|Interní identifikace (EKT0182 = BIC|IID).
C	MOKAS50	170	Hodnota údaje Identifikace zákazníka (EKT0005) není v souladu s vykázaným Typem identifikace zákazníka = IČO|NID|RČ (EKT0182 = IC|NID|RC).
C	MOKAS50	20	Hodnota údaje Identifikace účastníka (R0006) není v souladu s vykázaným Typem identifikace osoby (P0523).
C	MOKAS50	60	Hodnota údaje Identifikace investičního nástroje (EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	MOKAS60	20	Hodnota údaje Identifikace investičního nástroje (EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	MOKAS60	70	Hodnota údaje Identifikace podkladového nástroje (EKT0110) není v souladu s vykázaným Typem identifikace podkladového nástroje (EKT0111).
C	MOKAS61	20	Hodnota údaje Identifikace účastníka (R0006) není v souladu s vykázaným Typem identifikace osoby (P0523).
C	MOKAS61	40	Hodnota údaje Identifikace investičního nástroje (EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	MOKAS61	60	Hodnota údaje Protistrana (EKT0001) není v souladu s vykázaným Typem identifikace protistrany (EKT0181).
C	MOKAS61	80	Hodnota údaje Identifikace zákazníka (EKT0005) není v souladu s vykázaným Typem identifikace zákazníka (EKT0182).
C	MOKAS62	30	Hodnota údaje Identifikace investičního nástroje (EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	MOKAS63	20	Hodnota údaje Identifikace investičního nástroje

Třída	ID výkazu	ID kontroly	Název kontroly
			(EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	MOKAS65	20	Hodnota údaje Identifikace účastníka (R0006) není v souladu s vykázaným Typem identifikace osoby (P0523).
C	MOKAS70	20	Hodnota údaje Identifikace indexu (EKT0145) není v souladu s vykázaným Typem identifikace indexu (EKT0146).
C	MOKAS70	30	Hodnota údaje Identifikace investičního nástroje (EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
C	MOKAS80	110	Hodnota údaje Identifikace zákazníka (EKT0005) není v souladu s vykázaným Typem identifikace zákazníka = BIC|Interní identifikace (EKT0182 = BIC|IID).
C	MOKAS80	120	Hodnota údaje Identifikace zákazníka (EKT0005) není v souladu s vykázaným Typem identifikace zákazníka = Nula Nula IČO|IČO|NID|RČ (EKT0182 = 00IC|IC|NID|RC).
C	MOKAS80	50	Hodnota údaje Identifikace investičního nástroje (EKT0107) není v souladu s vykázaným Typem identifikace investičního nástroje (EKT0108).
D	JISIFE51	130	Pro hodnoty údaje Druh cenného papíru = dluhové (P0088 = 1%|311|312|36) musí být podíl hodnot údajů Množství / Nominální hodnota (BCP2005) (v Kč nebo po převodu na Kč) a Účetní hodnota (netto) (BCP2007) v intervalu 50 až 150%.
D	MOKAS42	10	Pro hodnotu údaje Typ transakce <> převod|derivát (P0504 <> 161|162|163|30|329|33) musí být hodnota údaje Jednotková cena (TKT0101|TKT0102) > 0.
D	MOKAS42	40	Pro hodnotu údaje Typ transakce <> převod (P0504 <> 161|162|163|329) má být hodnota údaje Jednotková cena v procentech (TKT0102) mezi 50 a 150. Kontrolují se jen řádky s dluhopisy.
D	MOKAS60	100	Pro hodnotu údaje Druh nástroje (CFI) = dluhopisy|akcie (SKT0004 = D%|E%) nesmí být vyplněn údaj Identifikace podkladového nástroje (EKT0110).
D	MOKAS60	110	Pro hodnotu údaje Druh nástroje (CFI) = akcie (SKT0004 = E%) nesmí být vyplněn údaj Datum splatnosti/realizace (DKT0007).
D	MOKAS60	120	Pro hodnotu údaje Druh nástroje (CFI) = dluhopisy|futures|opce|práva (SKT0004 = D%|F%|O%|R%) musí být vyplněn údaj Datum splatnosti/realizace (DKT0007).
D	MOKAS60	140	Pro hodnotu údaje Druh nástroje (CFI) = dluhopisy|akcie|futures (SKT0004 = D%|E%|F%) nesmí být vyplněn údaj Realizační cena (SKT0005).

Třída	ID výkazu	ID kontroly	Název kontroly
D	MOKAS60	150	Pro hodnotu údaje Druh nástroje (CFI) = opce|práva (SKT0004 = O%|R%) musí být vyplněn údaj Realizační cena (SKT0005).
D	MOKAS60	160	Pro hodnotu údaje Druh nástroje (CFI) = dluhopisy|akcie (SKT0004 = D%|E%) nesmí být vyplněn údaj Cenový multiplikátor/hodnota bodu (SKT0006).
D	MOKAS60	170	Pro hodnotu údaje Druh nástroje (CFI) = futures|opce|práva (SKT0004 = F%|O%|R%) musí být vyplněn údaj Cenový multiplikátor/hodnota bodu (SKT0006).
D	MOKAS60	180	Pro hodnotu údaje Druh nástroje (CFI) = dluhopisy (SKT0004 = D%) musí být vyplněn údaj Aktuální nominální hodnota (TKT0005).
D	MOKAS60	190	Pro hodnotu údaje Druh nástroje (CFI) = futures|opce|práva (SKT0004 = F%|O%|R%) nesmí být vyplněn údaj Aktuální nominální hodnota (TKT0005).
D	MOKAS60	210	Pro hodnotu údaje Druh nástroje (CFI) = akcie|futures|opce|práva (SKT0004 = E%|F%|O%|R%) nesmí být vyplněn údaj Země konečného vlastníka (EKT0013).
D	MOKAS60	220	Pro hodnotu údaje Druh nástroje (CFI) = dluhopisy (SKT0004 = D%) musí být vyplněn údaj Země konečného vlastníka (EKT0013).
D	MOKAS60	230	Pro hodnotu údaje Druh nástroje (CFI) = dluhopisy|akcie (SKT0004 = D%|E%) nesmí být vyplněn údaj Země indexu (EKT0014).
E	MOKAS40	130	Hodnota údaje Identifikace místa (EKT0002) není BIC ani MIC.
E	MOKAS43	70	Hodnota údaje Druh nástroje (CFI) (SKT0004) musí odpovídat ISO10962.
E	MOKAS44	40	Hodnota údaje Autor identifikace osoby (EKT0061) není BIC.
E	MOKAS44	60	Hodnota údaje Identifikace osoby (IČ) (EKT0062) není IČ.
E	MOKAS50	140	Hodnota údaje Identifikace místa (EKT0002) není BIC ani MIC.
E	MOKAS60	90	Hodnota údaje Druh nástroje (CFI) (SKT0004) musí odpovídat ISO10962.
E	MOKAS62	20	Hodnota údaje Identifikace účastníka (R0006) není BIC ani Interní identifikace na bázi MIC.
E	MOKAS80	100	Hodnota údaje Autor externího čísla transakce (EKT0029) není BIC ani MIC.
E	MOKAS80	130	Hodnota údaje Identifikace osoby vedoucí evidenci (EKT0034) není BIC ani Interní identifikace na bázi BIC.
E	MOKAS80	20	Hodnota údaje Identifikace účastníka (R0006) není BIC

Třída	ID výkazu	ID kontroly	Název kontroly
			ani Interní identifikace na bázi BIC.
E	MOKAS80	30	Hodnota údaje Identifikace strany transakce (EKT0026) není BIC ani Interní identifikace na bázi BIC.
E	MOKAS80	70	Hodnota údaje Identifikace účastníka - protistrany (EKT0027) není BIC ani Interní identifikace na bázi BIC.
E	MOKAS80	80	Hodnota údaje Protistrana (EKT0001) není BIC ani Interní identifikace na bázi BIC.
E	MOKAS80	90	Hodnota údaje Identifikace místa (EKT0002) není BIC ani MIC.
F	MOKAS40	10	Hodnota údaje Identifikace ohlašujícího subjektu (R0001) musí obsahovat BIC vykazujícího subjektu.
F	MOKAS40	150	Hodnota údaje Autor čísla obchodu (EKT0004) musí být BIC vykazujícího subjektu nebo MIC organizátora regulovaného trhu.
F	MOKAS40	20	Hodnota údaje Autor čísla hlášení (R0003) musí obsahovat BIC vykazujícího subjektu nebo organizátora regulovaného trhu.
F	MOKAS42	60	Pro hodnotu údaje Číslo pokynu (EKT0049) = VLASTNI musí údaj Identifikace zákazníka (EKT0005) obsahovat BIC vykazujícího subjektu.
F	MOKAS50	10	Kombinace hodnot údajů Hlášení OCP (OKT0004) = Y a Identifikace účastníka (R0006) <> BIC vykazujícího subjektu není povolena.
F	MOKAS60	30	Kombinace hodnot údaje Typ identifikace investičního nástroje (EKT0108) <> ISIN a Obchodní skupina (EKT0016) = regulovaný trh není povolena.
G	MOKAS40	40	Časová zóna v hodnotě údaje Čas uskutečnění obchodu (DKT0001+DKT0002) neodpovídá Česku.
G	MOKAS40	50	Hodnota údaje Den uskutečnění obchodu (DKT0001) je z budoucnosti.
G	MOKAS41	40	Hodnota údaje Přijetí pokynu - datum (DKT0026) je z budoucnosti.
G	MOKAS42	90	Hodnota údaje Den uskutečnění obchodu (DKT0001) je z budoucnosti.
G	MOKAS44	50	Hodnota údaje Datum narození (DKT0032) je z budoucnosti.
G	MOKAS50	40	Časová zóna v hodnotě údaje Čas uskutečnění obchodu (DKT0001+DKT0002) neodpovídá Česku.
G	MOKAS50	50	Hodnota údaje Den uskutečnění obchodu (DKT0001) je z budoucnosti.
G	MOKAS80	40	Časová zóna v hodnotě údaje Čas uskutečnění obchodu (DKT0001+DKT0002) neodpovídá Česku.
H	MOKAS41	250	K zaslanému hromadnému pokynu (hodnota údaje Identifikace zákazníka (EKT0005) = MORE CLIENTS) nebyly v rámci výskytu nalezeny dílčí pokyny (hodnota

Třída	ID výkazu	ID kontroly	Název kontroly
			údaje Identifikace zákazníka (EKT0005) <> MORE CLIENTS). Kontroluje se shoda prvních 11 znaků údaje Číslo pokynu (EKT0049).
H	MOKAS42	250	K zaslanému hromadnému pokynu (hodnota údaje Identifikace zákazníka (EKT0005) = MORE CLIENTS) nebyly v rámci výskytu nalezeny dílčí pokyny (hodnota údaje Identifikace zákazníka (EKT0005) <> MORE CLIENTS). Kontroluje se shoda prvních 11 znaků údaje Číslo pokynu (EKT0049).
I	JISIFE51	230	Rušený záznam nebyl nahlášen.
I	JISIFE52	70	Rušený záznam nebyl nahlášen.
I	JISIFE53	60	Rušený záznam nebyl nahlášen.
I	MOKAS40	210	Referenční číslo rušeného hlášení (R0004) Autora čísla rušeného hlášení (R0005) neexistuje.
I	MOKAS41	240	Rušený záznam nebyl nahlášen.
I	MOKAS42	140	Rušený záznam nebyl nahlášen.
I	MOKAS43	80	Rušený záznam nebyl nahlášen.
I	MOKAS44	80	Rušený záznam nebyl nahlášen.
I	MOKAS50	210	Referenční číslo rušeného hlášení (R0004) neexistuje.
J	MOKA41_52	10	Hodnota údaje Identifikace zadavatele (MOKAS41.EKT0051) nebyla nalezena v seznamu hodnot Identifikace osoby (MOKAS44.R0010).
J	MOKA41_52	20	Hodnota údaje Identifikace zákazníka (MOKAS41.EKT0005) nebyla nalezena v seznamu hodnot Identifikace osoby (MOKAS44.R0010).
J	MOKA41_52	50	Hodnota údaje Makléř 1 (MOKAS41.EKT0056) nebyla nalezena v seznamu hodnot Identifikace osoby (MOKAS44.R0010).
J	MOKA41_52	60	Hodnota údaje Makléř 2 (MOKAS41.EKT0057) nebyla nalezena v seznamu hodnot Identifikace osoby (MOKAS44.R0010).
J	MOKA42_52	50	Hodnota údaje Makléř 3 (MOKAS42.EKT0058) nebyla nalezena v seznamu hodnot Identifikace osoby (MOKAS44.R0010).
J	MOKA42_52	60	Hodnota údaje Makléř 4 (MOKAS42.EKT0059) nebyla nalezena v seznamu hodnot Identifikace osoby (MOKAS44.R0010).
K	MOKA41_51	10	Hodnota údaje Identifikace investičního nástroje (MOKAS41.EKT0107) nebyla nalezena ani v seznamu hodnot (MOKAS43.EKT0107) ani v seznamu akcií a dluhopisů přijatých k obchodování na regulovaných trzích EHP.
K	MOKA41_52	30	Hodnota údaje Požadovaná protistrana (MOKAS41.EKT0052) nebyla nalezena ani v seznamu hodnot Identifikace osoby (MOKAS44.R0010) ani v

Třída	ID výkazu	ID kontroly	Název kontroly
			seznamu mnohostranných obchodních systémů a regulovaných trhů EHP.
K	MOKA41_52	40	Hodnota údaje Identifikace třetí osoby (MOKAS41.EKT0054) nebyla nalezena ani v seznamu hodnot Identifikace osoby (MOKAS44.R0010) ani v seznamu mnohostranných obchodních systémů a regulovaných trhů EHP.
K	MOKA42_51	20	Hodnota údaje Identifikace investičního nástroje (MOKAS42.EKT0107) nebyla nalezena ani v seznamu hodnot (MOKAS43.EKT0107) ani v seznamu akcií a dluhopisů přijatých k obchodování na regulovaných trzích EHP.
K	MOKA42_52	10	Hodnota údaje Autor čísla obchodu (MOKAS42.EKT0004) nebyla nalezena ani v seznamu hodnot Identifikace osoby (MOKAS44.R0010) ani v seznamu mnohostranných obchodních systémů a regulovaných trhů EHP.
K	MOKA42_52	20	Hodnota údaje Identifikace zákazníka (MOKAS42.EKT0005) nebyla nalezena ani v seznamu hodnot Identifikace osoby (MOKAS44.R0010) ani v seznamu mnohostranných obchodních systémů a regulovaných trhů EHP.
K	MOKA42_52	30	Hodnota údaje Protistrana (MOKAS42.EKT0001) nebyla nalezena ani v seznamu hodnot Identifikace osoby (MOKAS44.R0010) ani v seznamu mnohostranných obchodních systémů a regulovaných trhů EHP.
K	MOKA42_52	40	Hodnota údaje Identifikace místa (MOKAS42.EKT0002) nebyla nalezena ani v seznamu hodnot Identifikace osoby (MOKAS44.R0010) ani v seznamu mnohostranných obchodních systémů a regulovaných trhů EHP.
L	MOKA42_53	10	Hodnota údaje Číslo pokynu (MOKAS42.EKT0049) nebyla nalezena v seznamu hodnot Číslo pokynu (MOKAS41.EKT0049) sestaveného za 6 měsíců zpětně vzhledem k hodnotě údaje Stav ke dni (MOKAS42.P0053).
L	MOKA42_53	20	Kombinace hodnot údajů Číslo pokynu (MOKAS42.EKT0049) a Identifikace zákazníka (MOKAS42.EKT0005) nebyla nalezena v seznamu hodnot Číslo pokynu (MOKAS41.EKT0049) a Identifikace zákazníka (MOKAS41.EKT0005) sestaveného za 6 měsíců zpětně vzhledem k hodnotě údaje Stav ke dni (MOKAS42.P0053).
L	MOKA42_53	30	Kombinace hodnot údajů Číslo pokynu (MOKAS42.EKT0049) a Identifikace investičního nástroje (MOKAS42.EKT0107) nebyla nalezena v seznamu hodnot Číslo pokynu (MOKAS41.EKT0049) a

Třída	ID výkazu	ID kontroly	Název kontroly
			Identifikace investičního nástroje (MOKAS41.EKT0107) sestaveného za 6 měsíců zpětně vzhledem k hodnotě údaje Stav ke dni (MOKAS42.P0053).
M	MOKA42_51	10	Pro hodnotu údaje Typ transakce <> převod|derivát (MOKAS42.P0504 <> 161|162|163|321|329|33) a Druh cenného papíru <> derivát (MOKAS43.P0088 < 4%) nesmí rozdíl mezi hodnotou údaje Objem obchodu (MOKAS42.TKT0104) a součinem hodnot údajů Množství/nominální hodnota (MOKAS42.TKT0103) a Jednotková cena (MOKAS42.TKT0101| MOKAS42.TKT0102/100) přesáhnout jeden dekadický řád.

Tabulka 7 - Popis kontrol v MKT

8.2 Příloha 2 - Příklady pro pravidla aktualizace a synchronizace Hierarchií číselníku

8.2.1 Pravidlo 1 — Vložení elementární Položky hierarchie do Hierarchie číselníku

Při vložení elementární Položky hierarchie do Hierarchie číselníku dochází ke změně množiny elementárních Položek hierarchie všech nadřízených uzlových Položek hierarchie vložené Položky hierarchie v dané Hierarchii číselníku. Tato změna se musí promítnout do všech souvisejících Hierarchii číselníku, kde se tyto uzlové Položky hierarchie vyskytují.

Příklad 1:

Hierarchie1 – původní stav	Hierarchie1 – po aktualizaci	Popis aktualizace
K1 U1 E1 E2 U2 E3 E4	K1 U1 E1 E2 U2 E3 E4 **E5**	Při aktualizaci Hierarchie1 byla vložena Položka hierarchie E5 pod uzlovou položku U2. Položka E5 se tak stává elementární položkou jak pro uzlovou položku U2, tak pro kořenovou položku K1.
Hierarchie2 – původní stav	**Hierarchie2 – po synchronizaci**	**Popis synchronizace**
K1 E1 E2 U2 U21 E3 E4	K1 E1 E2 U2 U21 E3 E4	Hierarchie2 obsahuje společnou uzlovou položku U2 a společnou kořenovou položku K1 s Hierarchií1 a bude synchronizována. Položka hierarchie E5 je vložena do Hierarchie2 na nejnižší

	E5	možnou úroveň společné uzlové položky U2 tj. na úroveň Položky hierarchie U21. Položka hierarchie E5 se stává elementární položkou pro uzlovou položku U2 a kořenovou položku K1.
Hierarchie3 – původní stav	**Hierarchie3 – po synchronizaci**	**Popis synchronizace**
K2 K1 E1 E2 E3 E4 U3 E6 E7	K2 K1 E1 E2 E3 E4 **E5** U3 E6 E7	Hierarchie3 obsahuje kořenovou položku K2, jež má podřízenou uzlovou položku K1. Položka hierarchie E5 je při synchronizaci vložena pod společnou uzlovou položku K1 s Hierarchií1. Položka hierarchie E5 se stává elementární položkou uzlové položky K1 a kořenovou položku K2.

Tabulka 8 - Vložení elementární Položky hierarchie do Hierarchie číselníku

8.2.2 Pravidlo 2 — Smazání elementární Položky hierarchie z Hierarchie číselníku

Při smazání elementární Položky hierarchie z Hierarchie číselníku dochází k jejímu smazání z množiny elementárních Položek hierarchie všech nadřízených uzlových Položek hierarchie v této Hierarchii číselníku, které v dané Hierarchii číselníku zůstaly. Tato změna se musí promítnout do všech souvisejících Hierarchií číselníku, kde se dané uzlové Položky hierarchie vyskytují.

Příklad 1:

Hierarchie1 – původní stav	**Hierarchie1 – po aktualizaci**	**Popis aktualizace**
K1 U1 E1 E2 U2 E3 E4 E5	K1 U1 E1 E2 U2 E3 E4 ~~**E5**~~	Při aktualizaci Hierarchie1 byla zrušena elementární položka E5 z uzlové položky U2. Zároveň položka E5 přestává být elementární položkou kořenové položky K1.
Hierarchie2 – původní stav	**Hierarchie2- po synchronizaci**	**Popis synchronizace**
K2	K2	Hierarchie2 obsahuje společnou

E1 E2 U2 E3 E4 E5 U3 E6 E7	E1 E2 U2 E3 E4 ~~E5~~ U3 E6 E7	uzlovou položku U2 s Hierarchií 1 a bude synchronizována. Položka E5 je zrušena z uzlové položky U2. Zároveň položka E5 přestává být elementární položkou i pro kořenovou položku K2.
Hierarchie3 – původní stav	**Hierarchie3 – po synchronizaci**	**Popis synchronizace**
K2 K1 E1 E2 E3 E4 E5 U3 E6 E7	K2 K1 E1 E2 E3 E4 ~~E5~~ U3 E6 E7	Hierarchie3 obsahuje společnou uzlovou položky K1 s Hierarchií1. Položka E5 je přímo pod K1, a protože uzel začínající uzlovou položkou K1 byl v Hierarchii1 modifikován, bude Hierarchie3 při synchronizaci modifikována také. Výsledkem je odstranění položky E5 z uzlové položky K1 a současně i z jejího nadřízené uzlové položky K2. Uzlové položka U3 zůstává nezměněna.

Tabulka 9 - Smazání elementární Položky hierarchie z Hierarchie číselníku

8.2.3 Pravidlo 3 — Smazání uzlové položky z Hierarchie číselníku

Při smazání uzlové položky (bez smazání jejího elementárního obsahu) z aktualizované Hierarchie číselníku nedochází ke změně množiny elementárních položek této uzlové položky a do ostatních Hierarchií číselníku se smazání uzlové položky nepromítá. Toto smazání je možno ignorovat.

Příklad 1: - situace, kdy je pouze smazána uzlová položka, ale její elementární obsah zůstává

Hierarchie1 – původní stav	**Hierarchie1 – po aktualizaci**	**Popis aktualizace**
K1 U1 E1 E2 U2 E3 E4 E5	K1 U1 E1 E2 ~~U2~~ E3 E4 E5	Při aktualizaci Hierarchie1 byla smazána uzlová položka U2 z kořene hierarchie K1. Elementární položky E3,E4,E5 zůstaly v Hierarchii1 a jejich přímou nadřízenou se stala kořenová položka K1, jinak nedošlo ke změně elementárního obsahu žádné uzlové položky. Tato aktualizace neovlivní

		synchronizaci souvisejících Hierarchií číselníku, protože uzlová položka U2 je smazána pouze z Hierarchie1, nadále ale zůstává uzlovou v ostatních Hierarchiích číselníku. Elementární obsah kořenové položky K1 aktualizací nebyl ovlivněn.
Hierarchie2 – původní stav	**Hierarchie2 – po synchronizaci**	**Popis synchronizace**
K2 E1 E2 U2 E3 E4 E5 U3 E6 E7	K2 E1 E2 U2 E3 E4 E5 U3 E6 E7	Beze změny. Uzlová položka U2 je zde metodiky požadována.

Tabulka 10 - Smazání uzlové položky z Hierarchie číselníku

8.2.4 Pravidlo 4 — Smazání uzlové položky z Hierarchie číselníku včetně jejího elementárního obsahu

Při smazání uzlové položky včetně jejího elementárního obsahu z aktualizované Hierarchie číselníku nedochází ke změně množiny elementárních položek této uzlové položky a do ostatních souvisejících Hierarchií číselníku se promítá změna elementárního obsahu v souvisejících uzlech Hierarchie číselníku. Při synchronizaci dojde ke sladění obsahu souvisejících uzlových položek.

Příklad 1: situace, kdy je smazán celý uzel začínající uzlovou položkou U2

Hierarchie1 – původní stav	**Hierarchie1 – po aktualizaci**	**Popis aktualizace**
K1 U1 E1 E2 U2 E3 E4 E5	K1 U1 E1 E2 ~~U2~~ ~~E3~~ ~~E4~~ ~~E5~~	Při aktualizaci Hierarchie1 byla smazána uzlová položka U2 včetně svého elementárního obsahu. Elementární obsah uzlové položky U2 se nezměnil, změnil se ovšem elementární obsah kořenové položky K1. Dále se uplatňuje pravidlo 2.

Hierarchie2 – původní stav	Hierarchie2 – po synchronizaci	Popis synchronizace
K2 K1 E1 E2 **E3** **E4** **E5** U3 E6 E7	K2 K1 E1 E2 ~~**E3**~~ ~~**E4**~~ ~~**E5**~~ U3 E6 E7	Hierarchie2 obsahuje společnou uzlovou položku K1 s Hierarchií 1. V Hierarchii2 bude synchronizován obsah společné uzlové položky K1, a to tak, že budou smazány elementární položky E3, E4, E5. Znamená to také, že úpravou bude dotčena i nadřízená kořenová položka K2. Elementární Obsah uzlové položky U3 zůstane nezměněn.

Tabulka 11 - Smazání uzlové položky z Hierarchie číselníku včetně jejího elementárního obsahu

8.2.5 Pravidlo 5 — Vložení uzlové položky do Hierarchie číselníku

Vložení uzlové položky v aktualizované Hierarchii číselníku pod jinou uzlovou položku se nepromítá do ostatních Hierarchií číselníku. Samotné toto vložení je možno ignorovat, musí však být provedena kontrola, zda neexistuje nějaká související Hierarchie číselníku, kde je vztah obou Položek hierarchie opačný (tzv. záměna uzlů). Pokud takový případ existuje, synchronizace se nepovolí.

Při vložení uzlové položky do aktualizované Hierarchie číselníku však může dojít ke změně množiny elementárních položek této uzlové položky. Tento rozdíl se musí zjistit (dále se uplatňují pravidla 1 a 2).

Příklad 1:

Hierarchie1 – původní stav	Hierarchie1 – po aktualizaci	Popis aktualizace
K1 U1 E1 E2 U2 E3 E4 E5	**K1** **U3** U1 E1 E2 U2 E3 E4 E5	Při aktualizaci Hierarchie1 byla vložena uzlová položka U3 pod kořenovou položku K1 a nad uzlové položky U1 a U2. V rámci synchronizace se musí provést kontrola, zda nedošlo k záměně uzlů. Položka U3 nebyla použita v žádném jiné Hierarchii číselníku (byla elementární položkou Číselníku). Po vložení do Hierarchie1 se změnil se její elementární obsah – vložení položek A1, A2, A3, B1, B2, B3 a položka se stala součtovou v Číselníku a

		uzlovou v Hierarchii číselníku. Pozn.: v číselníku pak systém nastaví položce U3 atribut „suma“ na „ano“.
Hierarchie2 – původní stav	**Hierarchie2 – po synchronizaci**	**Popis synchronizace**
K1 E1 E2 U2 U4 E3 U5 E4 E5	K1 E1 E2 U2 U4 E3 U5 E4 E5	Beze změny.

Tabulka 12 - Vložení uzlové položky do Hierarchie číselníku (Příklad 1)

Příklad 2 – ukázka konfliktu a nepovolené úpravy Hierarchie číselníku

Hierarchie1 – původní stav	**Hierarchie1 – po aktualizaci**	**Popis aktualizace**
K1 U1 E1 E2 U2 E3 E4 E5	K1 **U3** U1 E1 E2 U2 E3 E4 E5	Při aktualizaci byla vložena uzlová položka U3 pod kořenovou položku K1 a nad uzlové položky U1 a U2. V rámci synchronizace se musí provést kontrola, zda nedošlo k záměně uzlů. Položka U3 je použita v jiné hierarchii.
Hierarchie3 – původní stav	**Hierarchie3 – po synchronizaci**	**Popis synchronizace**
K3 K1 **E1** **E2** **E3** **E4** **E5** U3 E6 E7	K3 K1 E1 E2 E3 E4 E5 **U3** **KONFLIKT** E6 E7	Nastává konflikt, protože uzlová položka U3 v této hierarchii obsahuje jiný elementární obsah. I při sladění obsahu U3 dochází k porušení zásadního pravidla tvorby hierarchie a to že položka hierarchie smí být obsažena pouze jednou v hierarchii. Výsledek – synchronizace není povolena a končí chybou.

Tabulka 13 - Vložení uzlové položky do Hierarchie číselníku (Příklad 2)

Příklad 3:

Hierarchie1 – původní stav	Hierarchie1 – po aktualizaci	Popis aktualizace
K1 U1 E1 E2 U2 E3 E4 E5	K1 **U3** U1 E1 E2 U2 E3 E4 E5	Při aktualizaci byla vložena uzlová položka U3 pod kořenovou položku Hierarchie1 K1 a nad uzlové položky U1 a U2. V rámci synchronizace se musí provést kontrola, zda nedošlo k záměně uzlů. Položka U3 je použita také v Hierarchii2, kde je uzlová s elementárními položkami E1, E2, E3, E4,E5 a E6.
Hierarchie2 – původní stav	**Hierarchie2 – po synchronizaci**	**Popis synchronizace**
K2 U3 E1 E2 E3 E4 E5 E6	K2 U3 E1 E2 E3 E4 E5 ~~**E6**~~	Při synchronizaci dojde ke smazání elementární položky E6 z uzlové položky U3 v Hierarchii2. Změna elementárního obsahu promítne do kořenové položky K2.

Tabulka 14 - Vložení uzlové položky do Hierarchie číselníku (Příklad 3)

8.2.6 Pravidlo 6a — Změna uzlové položky na elementární položku

Ke změně uzlové položky na elementární může dojít pouze v rámci aktualizace smazáním všech jejích podřízených Položek hierarchie, nemůže vzniknout nová elementární položka z uzlové položky v synchronizované Hierarchii číselníku (souvisí s pravidlem 7).

Pokud se položka, která se stala v rámci aktualizace hierarchie elementární z uzlové položky, také vyskytuje v některé související Hierarchii číselníku, tak v této Hierarchii číselníku zůstane. Toto může způsobit sekundární vložení této položky do některé další uzlové položky.

Hierarchie1 – původní stav	Hierarchie 1 – po aktualizaci	Popis aktualizace
K1 U1 E1 E2 U2 E3	K1 U1 E1 E2 U2 ~~E3~~	Při aktualizaci byly smazány elementární položky E3, E4, E5, E6 z uzlové položky U2, která zůstala v Hierarchii1 a stala se tak elementární položkou Hierarchie číselníku.

E4 E5 E6	~~E4~~ ~~E5~~ ~~E6~~	Protože položka U2 je pod kořenovou položkou K1, byly elementární položky E3, E4, E5, E6 smazány i z K1. Výstupem aktualizace je zrušení elementárních položek E3, E4, E5, E6 z uzlových položek U2 a K1.
Hierarchie2 – původní stav	**Hierarchie2 – po synchronizaci**	**Popis synchronizace**
K1 E1 E2 U2 U3 E3 E4 U4 E5 E6	K1 E1 E2 U2 ~~U3~~ ~~E3~~ ~~E4~~ ~~U4~~ ~~E5~~ ~~E6~~	Hierarchie2 obsahuje uzlové položky U2 i K1 a bude synchronizována. Uzlová položka U2 obsahuje další úroveň uzlových položek U3 (E3, E4) a U4 (E5, E6). Položky E3, E4, E5, E6 budou ze synchronizované Hierarchie číselníku zrušeny i se svojí nadřízenou úrovní U3 a U4 (viz kapitola 8.2.8 Pravidlo 7 — Smazání celého uzle hierarchie včetně uzlové položky). Jelikož byly zrušeny uzlové položky U3 a U4, uplatní se pravidlo 3. Žádný další sekundární efekt zde proto nevzniká.
Hierarchie 3 – původní stav	**Hierarchie3 – po synchronizaci**	**Popis synchronizace**
K1 E1 E2 E3 E4 E5 E6	K1 E1 E2 U2 ~~E3~~ ~~E4~~ ~~E5~~ ~~E6~~	Hierarchie 3 obsahuje pouze kořenovou položku K1. Položky E3, E4, E5, E6 jsou přímo pod ní a při synchronizaci se zruší. Naopak se vloží položka U2, která se stala po synchronizaci elementární.

Tabulka 15 - Změna uzlové položky na elementární položku

8.2.7 Pravidlo 6b — Změna elementární položky na uzlovou položku

Ke změně elementární položky na uzlovou může dojít vložením Položek hierarchie pod jinou Položku hierarchie (tzn. vytvoření další úrovně hierarchie pod již existující Položkou hierarchie). Toto vložení ovlivní elementární úroveň souvisejících Hierarchií číselníku v souvisejících uzlech hierarchie.

Poznámka: pokud se z elementární položky stane uzlová (součtová), nastaví jí systém atribut suma na ano a v číselníku ji označí znakem ∑.

Hierarchie1 – původní stav	Hierarchie1 – po aktualizaci	Popis aktualizace
K1 U1 E1 E2 U2	K1 U1 E1 E2 U2 **E3** **E4** **E5**	Při aktualizaci byly vloženy pod U2 položky E3, E4, E5 a položka U2 se tak stala položkou uzlovou – byl vytvořen uzel. Výstupem aktualizace je vložení nových elementárních položek E3, E4, E5 pod položku U2 a tedy i její nadřízenou položku K1. Byl změněn elementární rozpad položky K1.
Hierarchie2 – původní stav	**Hierarchie2 – po synchronizaci**	**Popis synchronizace**
K1 E1 E2 U2	K1 E1 E2 U2 **E3** **E4** **E5**	Hierarchie2 obsahuje jednu společnou uzlovou položku K1 s Hierarchií 1 a bude synchronizována. Položka U2 se vložením dalších položek o úroveň níže stává položkou uzlovou.

Tabulka 16 - Změna elementární položky na uzlovou položku

8.2.8 Pravidlo 7 — Smazání celého uzle hierarchie včetně uzlové položky

V rámci synchronizace se smaže uzlová položka z Hierarchie číselníku pouze a právě v případě, pokud jsou smazány všechny její podřízené položky i položka sama a nemůže se stát elementární dle pravidel 6.

Příklad 1:

Hierarchie1 – původní stav	Hierarchie1 – po aktualizaci	Popis aktualizace
K1 U1 E1 E2 U2 E3 E4 E5	K1 U1 E1 E2 ~~**U2**~~ ~~**E3**~~ ~~**E4**~~ ~~**E5**~~	Cílem aktualizace je zrušení celého uzle U2, tj. uzlové položky U2 včetně jejího elementárního obsahu. Výstupem je změna elementárních položek kořenové položky K1, obsah uzlové položky U1 zůstává nezměněn.
Hierarchie2 – původní stav	**Hierarchie2 – po synchronizaci**	**Popis synchronizace**
K1	K1	Hierarchie2 obsahuje kořenovou položku K1 a

E1 E2 U2 E3 E4 E5	E1 E2 ~~U2~~ ~~E3~~ ~~E4~~ ~~E5~~	uzlovou položku U2, které jsou společné s Hierarchií1. Hierarchie2 bude synchronizována. Protože uzlová položka U2 je z Hierarchie1 smazána, musí být v rámci synchronizace změněn obsah Hierarchie2, a to tak že celý uzel U2 je smazán.

Tabulka 17 - Smazání celého uzle hierarchie včetně uzlové položky

8.2.9 Konflikt typu Duplicita v nezávislých uzlech

Příklad 1: duplicita v nezávislých uzlech

Hierarchie1 – původní stav	Hierarchie1 – po aktualizaci	Popis aktualizace
K1 U1 E1 E2 U2 E3 E4 E5	**K1** **U1** **E1** **E2** **U2** **E3** **E4** **E5** **E6**	V rámci aktualizace Hierarchie1 byla pod uzlovou položku U2 přidána položka E6.
Hierarchie2 – původní stav	**Hierarchie2 – po synchronizaci**	**Popis synchronizace**
K2 U1 E1 E2 U2 E3 E4 E5 U3 E6 E7	K2 U1 E1 E2 U2 E3 E4 E5 **E6** U3 **E6** E7 **KONFLIKT**	Hierarchie2 již obsahuje elementární položku E6 pod uzlem U3. Vzniká konflikt a Hierarchie číselníku nelze synchronizovat. Bylo porušeno pravidlo, že jedna položka smí být obsažena v Hierarchii číselníku pouze jedenkrát. Řešení: uživatel se musí rozhodnout, zda je požadovaná akce z věcného hlediska správná a upravit Hierarchii číselníku postupně. Jedno z řešení je v prvním kroku odebrat položku E6 z uzle U3 v Hierarchie2, provést aktualizaci a synchronizaci a následně položku U6 vložit pod U2 v Hierarchii 1 a znovu provést aktualizaci a synchronizaci.

Tabulka 18 - Konflikt typu Duplicita v nezávislých uzlech

8.2.10 Konflikt typu Duplicita v závislých uzlech

Příklad 2: duplicita v závislých uzlech

Hierarchie1 – původní stav	Hierarchie 1 – po aktualizaci	Popis aktualizace
K1 U1 E1 E2 U2 E3 E4 E5	K1 U1 E1 E2 **E6** U2 E3 E4 E5	V rámci aktualizace byla pod uzlovou položku U1 přidána elementární položka E6.
Hierarchie2 – původní stav	**Hierarchie2 – po synchronizaci**	**Popis synchronizace**
K2 U1 E1 E2 U2 E3 E4 E5 E6 E7	K2 U1 E1 E2 **E6** U2 E3 E4 E5 **E6** E7 **KONFLIKT**	Hierarchie2 obsahuje již elementární položku E6 pod uzlem K2. Vzniká konflikt a Hierarchie číselníku nelze synchronizovat. Bylo porušeno pravidlo, že jedna Položka hierarchie smí být obsažena v Hierarchii číselníku pouze jedenkrát.

Tabulka 19 - Konflikt typu Duplicita v závislých uzlech

8.3 Příloha 3 - Příklady pro pravidla aktualizace Domén číselníku

8.3.1 Pravidlo 1

Aktualizace se týká pouze Domén číselníku vytvořených z Hierarchie číselníku. Domén vytvořených vybráním položek přímo z Číselníku se aktualizace nedotkne.

Hierarchie 1- původní stav	Hierarchie 1 -po aktualizaci	Popis aktualizace
K1 U1 E1 E2 E3 U2	K1 U1 E1 E2 E3 **E4**	Do Hierarchie1 byla vložena položka E4

F1 F2 F3	U2 F1 F2 F3	
Doména1 - původní stav	**Doména1 – stav po aktualizaci**	**Popis aktualizace**
K1 ∑ U1 ∑ E1 E2 E3 U2 ∑ F1 F2 F3	K1 ∑ U1 ∑ E1 E2 E3 **E4** U2 ∑ F1 F2 F3	Z Hierarchie byla vytvořena ruční Doména1, která kopíruje celou Hierarchii číselníku. Tato Doména číselníku je úplná ve všech uzlových položkách K1, U1, U2. Vložení položky E4 do hierarchie se projeví vložením položky E4 pod položku U1 (viz kapitola 3.9 Objekt Doména číselníku bod 8)
Doména2 - původní stav	**Doména2 – stav po aktualizaci**	**Popis aktualizace**
K1 U1 F1 F2 F3	K1 U1 F1 F2 F3 Beze změny	Doména2 je vytvořena přímo vybráním položek z číselníku. Změna v hierarchii se domény nedotkne.

Tabulka 20 – Aktualizace Domén číselníku – Pravidlo 1

8.3.2 Pravidlo 2

Aktualizace se týká pouze úplných uzlů (∑) u Domén číselníku vytvořených z Hierarchie číselníku. Neúplný uzel zůstává změnou v Hierarchii číselníku nedotčen.

8.3.3 Pravidlo 3

Pokud je smazána nebo vkládána Položka hierarchie z/do Hierarchie číselníku projeví se tato změna i v Doméně aktualizované s Hierarchií číselníku. Pokud je smazána nebo vkládána elementární položka do Hierarchie číselníku, pak se tato změna projeví pouze v úplném uzlu Domény vytvořené u Hierarchie číselníku.

Příklad 1

Hierarchie- původní stav	Hierarchie -po aktualizaci	Popis aktualizace

K1 U1 E1 E2 E3 U2 F1 F2 F3	K1 U1 E1 E2 E3 E4 U2 F1 F2 F3 F4	Do hierarchie byly přidány elementární položky hierarchie E4 pod uzlovou položku U1 a elementární položka hierarchie F4 pod uzlovou položku F4.
Doména1 - původní stav	**Doména1 – stav po aktualizaci**	**Popis aktualizace**
K1 ∑ U1 ∑ E1 E2 E3 U2 F1 F2	K1 ∑ U1 ∑ E1 E2 E3 E4 U2 F1 F2	Doména1 je úplná v uzlech K1 a U1. Změna přidání položky E4 se v doméně projeví přidáním položky E4 pod položku U1. Vložení položky F4 se neprojeví i když uzel K1 je úplný. Pozn.: Uzel K1 je úplný na úrovni položek U1 a U2.
Doména2 - původní stav	**Doména2 – stav po aktualizaci**	**Popis aktualizace**
K1 U1 ∑ E1 E2 E3 F1 F2	K1 U1 ∑ E1 E2 E3 E4 F1 F2	Doména 2 je úplná v uzlu U1. Změna v hierarchii se jí dotkne jen v tomto uzlu a to vložením položky E4 pod položku U1. Vložení položky F4 do hierarchie se v Doméně2 neprojeví, protože v doméně není obsažena položka U2 a kořenová položka K1 je neúplná.

Tabulka 21 - Aktualizace Domén číselníku – Pravidlo 3

8.3.4 Pravidlo 4

Pokud je smazána uzlová položka Hierarchie číselníku a její podřízené Položky hierarchie zůstávají, je změna promítnuta do Domény číselníku tak že, zůstávají jen elementární položky, aby se elementární obsah nadřízeného úplného uzlu nezměnil.

Hierarchie- původní stav	**Hierarchie -po aktualizaci**	**Popis aktualizace**
K1 U1 E1	K1 U1 E1	Z Hierarchie číselníku byla zrušena uzlová položka U2. Elementární položky F1, F2,

E2 E3 U2 F1 F2 F3	E2 E3 ~~U2~~ F1 F2 F3	F3 se posunou na úroveň položky U1 a, to proto, aby elementární obsah uzlové položky U1 zůstal stejný.
Doména1 - původní stav	**Doména1 – stav po aktualizaci**	**Popis aktualizace**
K1 ∑ U1 ∑ E1 E2 E3 U2 ∑ F1 F2 F3	K1 ∑ U1 ∑ E1 E2 E3 ~~U2~~ F1 F2 F3 Aktualizace Domén číselníku – Pravidlo 4	Doména 1 je úplná v uzlech K1, U1 a U2. Změna odebrání uzlové položky U2 se projeví v doméně tak, že je zrušen uzel U2 a zůstávají pouze elementární položky F1, F2 a F3 na úrovni součtové položky U1, jejíž obsah se úpravou Hierarchie číselníku neměnil.

Tabulka 22 - Aktualizace Domén číselníku – Pravidlo 4

8.4 Příloha 4 — Počty instancí vybraných objektů metapopisu v roce 2016 v systému MtS

Objekt	Počet	Poznámka
Výkaz	247	
Datová oblast	1144	
Číselník	259	
Položka číselníku	11377	Bez vykazovacích povinností.
Hierarchie číselníku	370	
Doména číselníku	4014	Včetně automatických domén vzniklých z Hierarchie číselníku a bez domén potřebných pro vykazovací povinnosti.
Převodník	2	
Účtová osnova	2	
Účet	499	
Datový typ	177	
Doména datového typu	60	
Ukazatel	9491	
Parametr	395	Bez vykazovacích povinností a technologických parametrů.
Jednovýkazová kontrola	15758	
Mezivýkazová kontrola	2394	
Skupina MVK	182	
Kontrola v časové řadě	498	Jen odchýlení od lineární regresní přímky.

Tabulka 23 - Počty vybraných objektů v systému MTS