

# Věcné zadání projektu SDAT

---

## B - Metapopis

## Obsah

1	Úvod.....	7
2	Obecný úvod do metapopisu.....	7
2.1	Údaj jako základní prvek metapopisu .....	7
2.2	Sledování historie instancí objektů metapopisu .....	9
2.2.1	Číslo verze a varianty instance objektu.....	9
2.2.2	Objektový model pro sledování historie instancí objektů.....	9
2.2.3	Přístup „Bez sledování historie“ .....	9
2.2.4	Přístup „Bez sledování historie – stavy“ .....	9
2.2.5	Přístup „Sledování historie – časová platnost“ .....	9
2.2.6	Přístup „Sledování historie – časová platnost + stavy“ .....	9
2.2.7	Přístup „Sledování historie – časová platnost na každém atributu“ .....	10
2.3	Vazby mezi jednotlivými objekty.....	10
2.3.1	Objekt #Objekt .....	11
2.3.2	Objekt #ObjektAtribut.....	11
2.3.3	Objekt #ObjektZávislost .....	12
2.3.4	Dopad editace instance objektu na číslo verze a varianty.....	14
2.3.5	Objektový model .....	16
2.3.6	Rámcové vymezení závislosti objektů .....	16
2.4	Objekty metapopisu .....	18
2.4.1	Základní vlastnosti objektů .....	19
2.4.2	Standardní atributy objektů .....	19
2.4.3	Základní principy práce s objekty .....	21
2.4.4	Životní cyklus objektu.....	22
2.4.5	Základní pravidla práce s instancemi objektů .....	24
2.5	Kontrola konzistence .....	26
3	Objektový model.....	28
3.1	Objekt Pracovní oblast.....	28
3.1.1	Atributy objektu Pracovní oblast.....	28
3.2	Objekt Výkaz.....	29
3.3	Objekt Blok výkazu .....	30
3.4	Objekt Datová oblast .....	31

3.4.1	Atributy objektu Datová oblast .....	34
3.5	Objekt Číselník .....	34
3.5.1	Atributy objektu Číselník .....	37
3.6	Objekt Položka číselníku .....	37
3.6.1	Atributy objektu Položka číselníku .....	38
3.6.2	Dynamické atributy objektu Položka číselníku.....	38
3.7	Objekt Hierarchie číselníku .....	40
3.7.1	Atributy objektu Hierarchie číselníku .....	42
3.7.2	Proces tvorby Hierarchií číselníku .....	42
3.8	Objekt Položka hierarchie .....	46
3.8.1	Atributy objektu Položka hierarchie .....	47
3.9	Objekt Doména číselníku .....	47
3.9.1	Atributy objektu Doména číselníku .....	50
3.10	Objekt Převodník.....	50
3.10.1	Atributy objektu Převodník .....	51
3.11	Objekt Položka převodníku .....	51
3.12	Objekt Účtová osnova .....	52
3.12.1	Atributy objektu Účtová osnova .....	53
3.13	Objekt Účet.....	53
3.13.1	Atributy objektu Účet.....	55
3.14	Objekt Datový typ .....	55
3.14.1	Atributy objektu Datový typ .....	56
3.15	Objekt Doména datového typu .....	56
3.15.1	Atributy objektu Doména datového typu.....	57
3.16	Objekt Ukazatel .....	57
3.16.1	Atributy objektu Ukazatel .....	59
3.17	Objekt Dodatečná konkretizace ukazatele.....	59
3.18	Objekt Parametr .....	59
3.18.1	Atributy objektu Parametr .....	61
3.19	Objekt Údaj.....	61
3.19.1	Vznik Údaje – statická Datová oblast .....	62
3.19.2	Popis Údaje nedimenzionálním parametrem.....	66
3.19.3	Dodatečná konkretizace ukazatele (nespolečné Parametry) .....	69

3.19.4	Sumarizační (neelementární) položky Hierarchie číselníku .....	71
3.19.5	Atributy objektu Údaj.....	72
3.20	Objekt Kontrola .....	73
3.20.1	Atributy objektu Kontrola .....	74
3.20.2	Způsob vytvoření kontroly .....	75
3.20.3	Objekt Jednovýkazová kontrola (JVK) .....	76
3.20.4	Objekt Kontrola časové řady (KČŘ).....	78
3.20.5	Objekt Mezivýkazová kontrola (MVK) .....	78
3.20.6	Odchylka v sémantických kontrolách .....	79
3.21	Objektový model – Definice kontrol .....	82
3.22	Objektový model – Metapopis .....	83
4	Uživatelské rozhraní .....	84
4.1	Knihovna .....	84
4.2	Příprava výkazů .....	84
4.3	Úložiště schválených a platných výkazů .....	85
5	Hlavní procesy .....	85
5.1	Proces tvorby metapopisu.....	85
5.1.1	Spouštěč procesu .....	86
5.1.2	Průběh procesu .....	86
5.1.3	Výstup procesu .....	87
5.2	Proces definice Pracovní oblasti.....	87
5.2.1	Spouštěč procesu .....	88
5.2.2	Průběh procesu .....	88
5.2.3	Výstup procesu .....	90
5.3	Proces tvorby Výkazu.....	91
5.3.1	Spouštěč procesu .....	93
5.3.2	Průběh procesu .....	93
5.3.3	Výstup procesu .....	99
5.4	Proces tvorby objektů popisujících údaje.....	99
5.4.1	Spouštěč procesu .....	100
5.4.2	Průběh procesu .....	100
5.4.3	Výstup procesu .....	105
5.5	Proces tvorba kontrol výkazu .....	105

5.5.1	Spouštěč procesu .....	107
5.5.2	Průběh procesu .....	107
5.5.3	Výstup procesu .....	110
5.6	Proces schválení výkazů v Pracovní oblasti .....	110
5.6.1	Spouštěč procesu .....	111
5.6.2	Průběh procesu .....	111
5.6.3	Výstup procesu .....	112
5.7	Proces prezentace Výkazu .....	112
5.7.1	Spouštěč procesu .....	113
5.7.2	Průběh procesu .....	113
5.7.3	Výstup procesu .....	116
6	Podpůrné procesy .....	116
6.1	Proces Přebírání metapopisu z externích zdrojů.....	116
6.1.1	Spouštěč procesu .....	117
6.1.2	Průběh procesu .....	117
6.1.3	Výstup procesu .....	118
6.1.4	Převodník výrazů Data point modelu (DPM) .....	118
6.2	Proces Navazování časových řad Údajů.....	119
6.2.1	Spouštěč procesu .....	120
6.2.2	Průběh procesu .....	120
6.2.3	Výstup procesu .....	121
7	Funkční požadavky .....	122
7.1	Obecné požadavky pro objekty metapopisu .....	122
7.2	Pracovní oblast .....	127
7.3	Výkaz.....	131
7.4	Blok výkazu.....	139
7.5	Datová oblast .....	145
7.6	Číselník, Položka číselníku.....	154
7.7	Hierarchie číselníku, Položka hierarchie .....	161
7.8	Doména číselníku .....	170
7.9	Převodník, Položka převodníku.....	175
7.10	Účtová osnova, účet.....	179
7.11	Datový typ .....	183

7.12	Doména datového typu.....	186
7.13	Ukazatel, Konkretizace ukazatele, Dodatečná konkretizace ukazatele.....	188
7.14	Parametr, Konkretizovaný parametr.....	198
7.15	Údaj .....	204
7.16	Jednovýkazová kontrola (JVK) .....	207
7.17	Mezivýkazová kontrola (MVK) .....	215
7.18	Kontrola časové řady (KČŘ).....	223
7.19	Knihovna .....	227
7.20	Navazování časových řad .....	229
8	Přílohy.....	230
8.1	Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systémem SDAT 230	
8.1.1	Kontroly v časové řadě (KČŘ).....	230
8.2	Příloha 2 - Příklady pro pravidla aktualizace a synchronizace Hierarchií číselníku 231	
8.2.1	Pravidlo 1 — Vložení elementární Položky hierarchie do Hierarchie číselníku 231	
8.2.2	Pravidlo 2 — Smazání elementární Položky hierarchie z Hierarchie číselníku 233	
8.2.3	Pravidlo 3 — Smazání uzlové položky z Hierarchie číselníku.....	234
8.2.4	Pravidlo 4 — Smazání uzlové položky z Hierarchie číselníku včetně jejího elementárního obsahu .....	235
8.2.5	Pravidlo 5 — Vložení uzlové položky do Hierarchie číselníku.....	236
8.2.6	Pravidlo 6a — Změna uzlové položky na elementární položku .....	239
8.2.7	Pravidlo 6b — Změna elementární položky na uzlovou položku.....	240
8.2.8	Pravidlo 7 — Smazání celého uzlu hierarchie včetně uzlové položky .....	241
8.2.9	Konflikt typu Duplicita v nezávislých uzlech .....	242
8.2.10	Konflikt typu Duplicita v závislých uzlech.....	243
8.3	Příloha 3 - Příklady pro pravidla aktualizace Domén číselníku .....	244
8.3.1	Pravidlo 1 .....	244
8.3.2	Pravidlo 2 .....	245
8.3.3	Pravidlo 3 .....	246
8.3.4	Pravidlo 4 .....	247
8.4	Příloha 4 — Počty instancí vybraných objektů metapopisu v roce 2015 v systému MtS 248	

## 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 vykážou 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 popsáný 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 dva typy Údajů, tj. statický a dynamický, které se od sebe liší rozsahem oboru hodnot parametrů:

#### a) statický Údaj

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

		Všechny měny	Koruna česká		Euro		Ostatní měny bez CZK a EUR	
		Rezidenti a nerezidenti celkem	Rezidenti	Nerezidenti	Rezidenti	Nerezidenti	Rezidenti	Nerezidenti
A	B	1	2	3	4	5	6	7
Aktiva	1					U1		
Pokladna	2			X	X		X	
Poskytnuté úvěry a vklady	3			U2				
Neobchodovatelné dluhové cenné papíry	4							
Ostatní aktiva bilanční	5							

**údaj U2** je definován ukazatelem *Poskytnuté úvěry a vklady*, konkretizovanými dimenzionálními parametry *Měna. Koruna česká, Sektor. Nerezidenti* a parametry *Jednotka vykazování. Koruna česká, Ocenění aktiv/pasív. Statistická nominální hodnota a Rozsah vykazování. ČR*

**údaj U1** je definován ukazatelem *Aktiva*, konkretizovanými dimenzionálními parametry *Měna. Euro, Sektor. Nerezidenti* a parametry *Jednotka vykazování. Koruna česká a Rozsah vykazování. ČR*

		Všechny měny	Koruna česká		Euro		Ostatní měny bez CZK a EUR	
		Rezidenti a nerezidenti celkem	Rezidenti	Nerezidenti	Rezidenti	Nerezidenti	Rezidenti	Nerezidenti
A	B	1	2	3	4	5	6	7
Aktiva bilanční celkem	1	360	30	80		200	50	
Pokladna	2			X	X		X	
Poskytnuté úvěry a vklady	3	80		80				
Neobchodovatelné dluhové cenné papíry	4	80	30					
Ostatní aktiva bilanční	5	200				200	50	

**hodnota údaje U2 za Vykazující osobu .XXXL banka a Stav ke dni 31.12.2012 je 80.**

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.

		Objem hypotečních úvěrů			Objem spotřebitelských úvěrů		
		Celkem	do splatnosti	po splatnosti	Celkem	do splatnosti	po splatnosti
A	B	1	2	3	4	5	6
Součet		1					
Výčet zemí	Výčet měn	2		U3			

**"neúplný" dynamický údaj U3** je definován ukazatelem *Objem hypotečních úvěrů*, konkretizovanými dimenzionálními parametry *Splatnost. po splatnosti, Země. Množina definovaných zemí a Měna. Množina definovaných měn a parametry Jednotka vykazování. Koruna česká a Rozsah vykazování. ČR*

		Objem hypotečních úvěrů			Objem spotřebitelských úvěrů		
		Celkem	do splatnosti	po splatnosti	Celkem	do splatnosti	po splatnosti
A	B	1	2	3	4	5	6
Součet		1	60000	58000	2000	20000	10000
ČR	Kč	2	2400	1900	500	300	300
Polsko	EUR	3	50	50			
Itálie	EUR	4	250		250	1100	400
Itálie	Dolar	5	300	50			
.....	.....	...					

**údaj je definován ukazatelem *Objem hypotečních úvěrů*, konkretizovanými dimenzionálními parametry *Splatnost. po splatnosti, Země. Itálie a Měna. EUR* a parametry *Jednotka vykazování. Koruna česká a Rozsah vykazování. ČR. Hodnota údaje za Vykazující osobu .XXXL banka a Stav ke dni 31.12.2012 je 250.***

Obrázek 2 - Znázornění dynamického Údaje a Hodnoty údaje



## 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 omezena, 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 navrhně 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 metaobjekty, 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 povolena.

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

- změna hodnoty atributu není povolena<sup>1</sup>,
- **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“,

<sup>1</sup> 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.

- **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 změna**“,
- **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“),
- 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 nadřazenou 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 nadřazeným). To znamená, že definuje, co se stane se souvisejícími nadřazenými 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 nadřazenému 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 nadřazených 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).

1.1.2013	31.12.4000
Výkaz V1_1	
1.1.2013	31.12.4000
Blok výkazu BV1_1	
1.1.2013	31.12.4000
Datová oblast DO1_1	
1.1.2013	31.12.4000
Datová oblast DO2_1	

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.

1.1.2013	31.12.2013	1.1.2014	31.12.4000
Výkaz V1_1		Výkaz V1_2	
1.1.2013			31.12.4000
Blok výkazu BV1_1		Blok výkazu BV1_2	
1.1.2013	31.12.2013	1.1.2014	31.12.4000
Datová oblast DO1_1		Datová oblast DO1_2	
1.1.2013			31.12.4000
Datová oblast DO2_1			

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 nadřazeného 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 nadřazeného objektu, ze které vznikla nová verze/varianta instance objektu.

Aby tento aparát fungoval, musejí být splněna následující pravidla:

- 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),
- 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,
- jeden Primární objekt může mít N Závislých objektů,
- 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

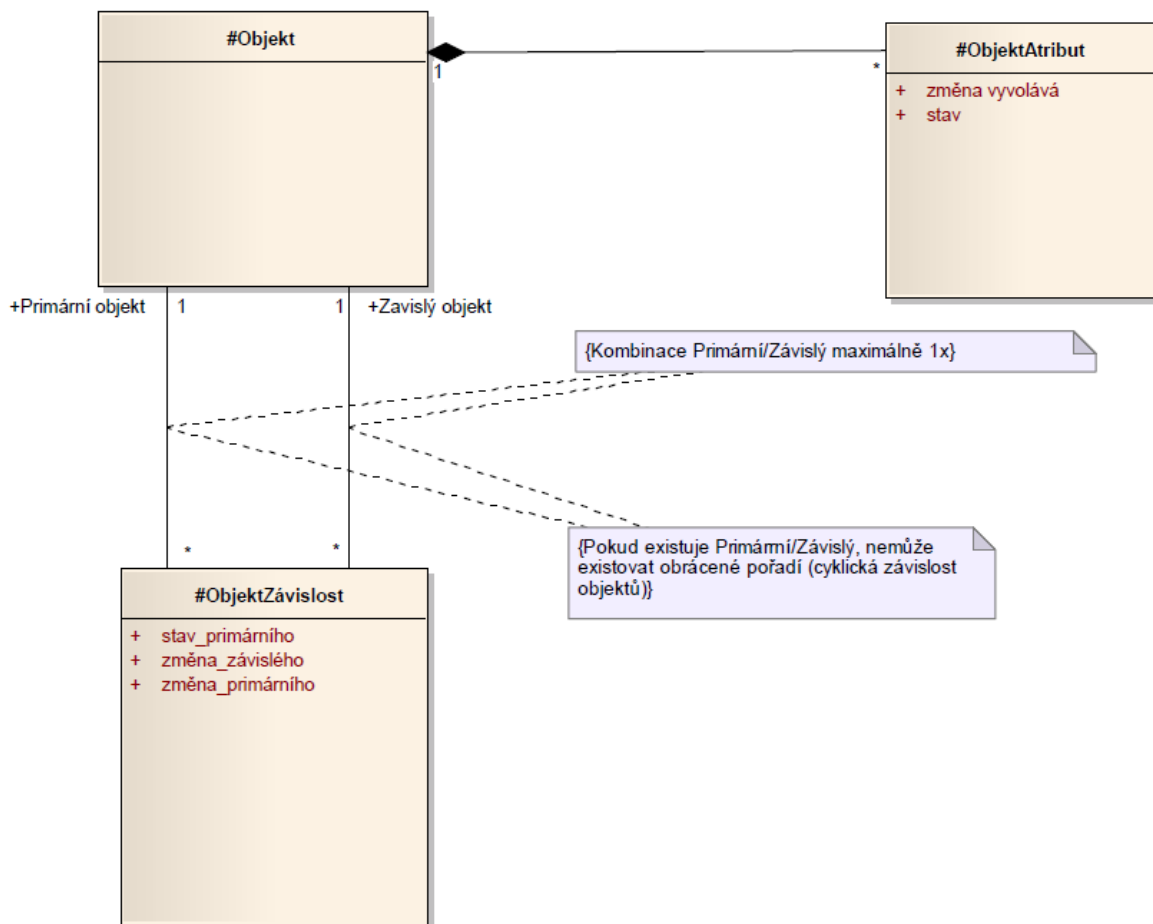
V případě, že se uživatel rozhodne měnit nějakou existující instanci jakéhokoli objektu metapopisu, postupuje se takto:

- 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).
- Uživatel se rozhodne, jaký typ změny chce provést, k dispozici má volby:
  - vytvořit novou verzi instance objektu,
  - vytvořit novou variantu instance objektu,
  - 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 změnit 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 attributech 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 závislosti 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 popsána 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 B 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	<ul style="list-style-type: none"> <li>Nová verze Výkazu</li> </ul>
Nová varianta Bloku	<ul style="list-style-type: none"> <li>Nová varianta Výkazu</li> </ul>
Nová verze Datové oblasti	<ul style="list-style-type: none"> <li>Nová verze Bloku</li> </ul>
Nová varianta Datové oblasti	<ul style="list-style-type: none"> <li>Nová varianta Bloku</li> </ul>
Nová verze Hierarchie číselníku	<ul style="list-style-type: none"> <li>Nová verze Datové oblasti, v jejíž konkretizaci Parametru je použita</li> <li>Nová verze Datové oblasti, v jejíž konkretizaci Ukazatele je použita (Konkretizovaný ukazatel, Dodatečná konkretizace ukazatele)</li> <li>Nová verze kontroly (JVK, MVK)</li> </ul>
Změna obsahu Hierarchie číselníku	<ul style="list-style-type: none"> <li>Nová verze příslušných Domén číselníku, které jsou modifikovány</li> <li>Nová verze příslušných Hierarchií číselníků, které jsou při synchronizaci modifikovány</li> </ul>
Nová verze Domény číselníku	<ul style="list-style-type: none"> <li>Nová verze Datové oblasti, v jejíž konkretizaci Parametru je použita</li> <li>Nová verze Datové oblasti, v jejíž konkretizaci Ukazatele je použita (Konkretizovaný Ukazatel, Dodatečná konkretizace ukazatele)</li> <li>Nová verze kontroly (JVK, MVK)</li> </ul>
Nová verze Parametru	<ul style="list-style-type: none"> <li>Nová verze Datové oblasti, v jejíž konkretizaci je Parametr</li> <li>Nová verze Datové oblasti, v jehož konkretizaci Ukazatele je použita (Dodatečná konkretizace ukazatele)</li> </ul>
Nová varianta Parametru	<ul style="list-style-type: none"> <li>Nová varianta Datové oblasti, v jejíž konkretizaci je Parametr</li> <li>Nová varianta Datové oblasti, v jejíž konkretizaci Ukazatele je použita (Dodatečná konkretizace ukazatele)</li> </ul>
Nová verze Datového typu	<ul style="list-style-type: none"> <li>Nová verze Datové oblasti, v jejíž konkretizaci Parametru je použita</li> <li>Nová verze Datové oblasti, v jejíž konkretizaci Ukazatele je použita (Konkretizovaný ukazatel, Dodatečná konkretizace ukazatele)</li> <li>Nová verze Kontroly</li> </ul>
Nová verze Domény datového typu	<ul style="list-style-type: none"> <li>Nová verze Datové oblasti, v jejíž konkretizaci Parametru je použita</li> <li>Nová verze Kontroly</li> <li>Nová verze Datové oblasti, v jejíž konkretizaci Ukazatele je použita (Konkretizovaný ukazatel, Dodatečná</li> </ul>

	konkretizace ukazatele)
Nová verze Ukazatele	<ul style="list-style-type: none"> <li>Nová verze Datové oblasti</li> </ul>
Nová varianta Ukazatele	<ul style="list-style-type: none"> <li>Nová varianta Datové oblasti</li> </ul>
Založení nového Účtu (tj. změna obsahu Účtové osnovy)	<ul style="list-style-type: none"> <li>Nová verze Ukazatele, je-li vazba Ukazatele na Účtovou osnovu definována celou skupinou nebo třídou, do níž byl nový Účet přidán</li> </ul>
Ukončení platnosti Účtu (tj. změna obsahu Účtové osnovy)	<ul style="list-style-type: none"> <li>Nová verze Ukazatele, kde byl Účet přiřazen</li> </ul>
Nová verze/varianta kontroly	<ul style="list-style-type: none"> <li>Nová varianta Výkazu</li> </ul>

**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 nadřizovaného objektu a změna podřizovaného objektu vyžaduje novou verzi nadřizovaného objektu, dochází k nahrazení stávající varianty nadřizovaného 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 nadřizován i podřizován, záleží na tom, jak je objekt použit (např. Doména číselníku, která je použita při definování obsahu dimenzionálního Parametru, je v tomto případě podřizován objektem. V jiném případě, kdy Doména číselníku je definována Položkami číselníku, je pak Doména nadřizován objektem).

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

Nadřizovaný objekt	Podřizovaný objekt		
	Projektovaný	Schválený	Platný
Projektovaný	Ano	Ano	Ano
Schválený	Ne	Ano	Ano
Platný	Ne	Ne	Ano

**Tabulka 2 - Podřizované a nadřizované 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 nadřazených objektech ve zvoleném časovém kontextu. Uživatel volí časový řez a má možnost zobrazit použití v tomto časovém řezu nebo v budoucím období, tj. v nadřazených 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. Uživateli 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 podtržítka),
- 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 nemění 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 nemění 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 nemění 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 systémem 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 platnosti Pracovní oblasti, nebo individuálně v případě, že platnost objektů je rozdílná od platnosti celé Pracovní oblasti. 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 verzi nebo variantou následuje další verze nebo varianta, systém přiřazuje platnost\_do tak, aby období platnosti plynule za sebou následovala. V případě ukončení platnosti objektu bez dalšího pokračování hodnotu platnost\_do přiřazuje uživatel. Poslední verze nebo varianta, pokud objektu není ukončena platnost, má hodnotu nastavenou 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 3.8.1 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 nadřazených 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.

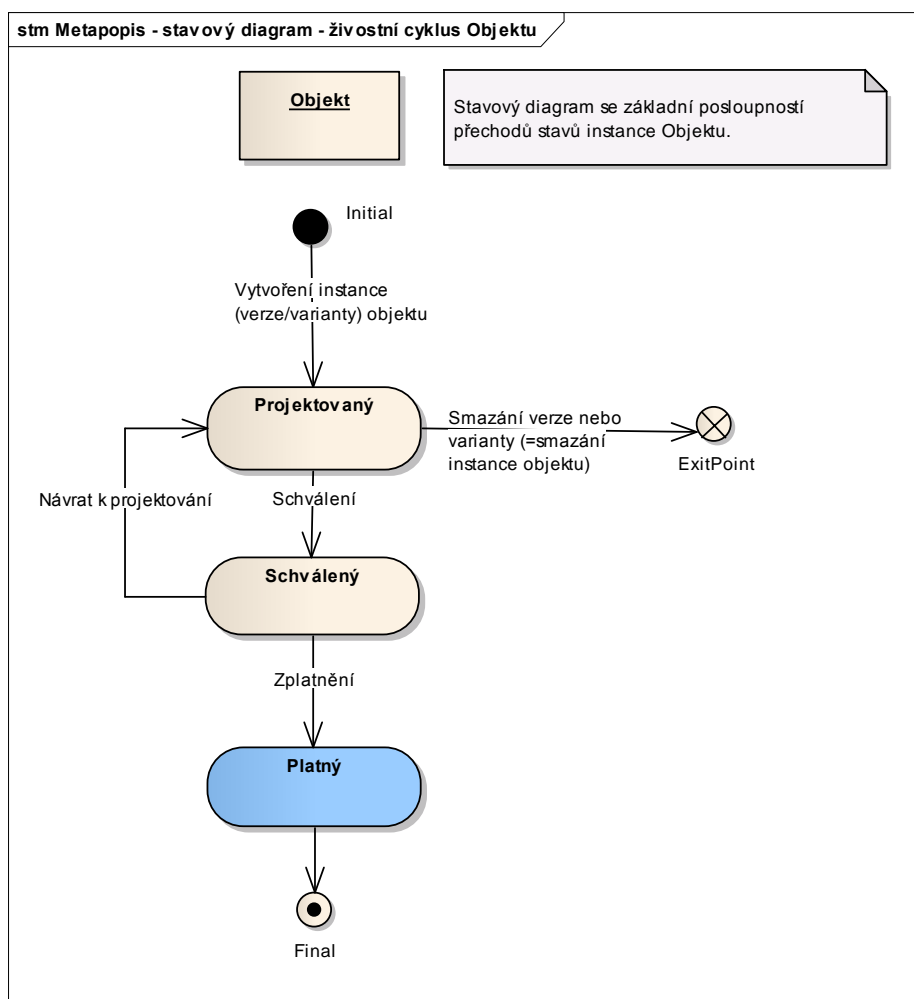
Pro všechny objekty platí základní pravidla pro práci s nimi, která se odvíjejí od stavu objektu, tj. musí platit následující:

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 neměnicí charakter objektu
Platný	Od data platnost_od objektu neomezeno	Platný	Ne = nutné vytvořit novou verzi	Ano, pouze atributy neměnicí charakter objektu

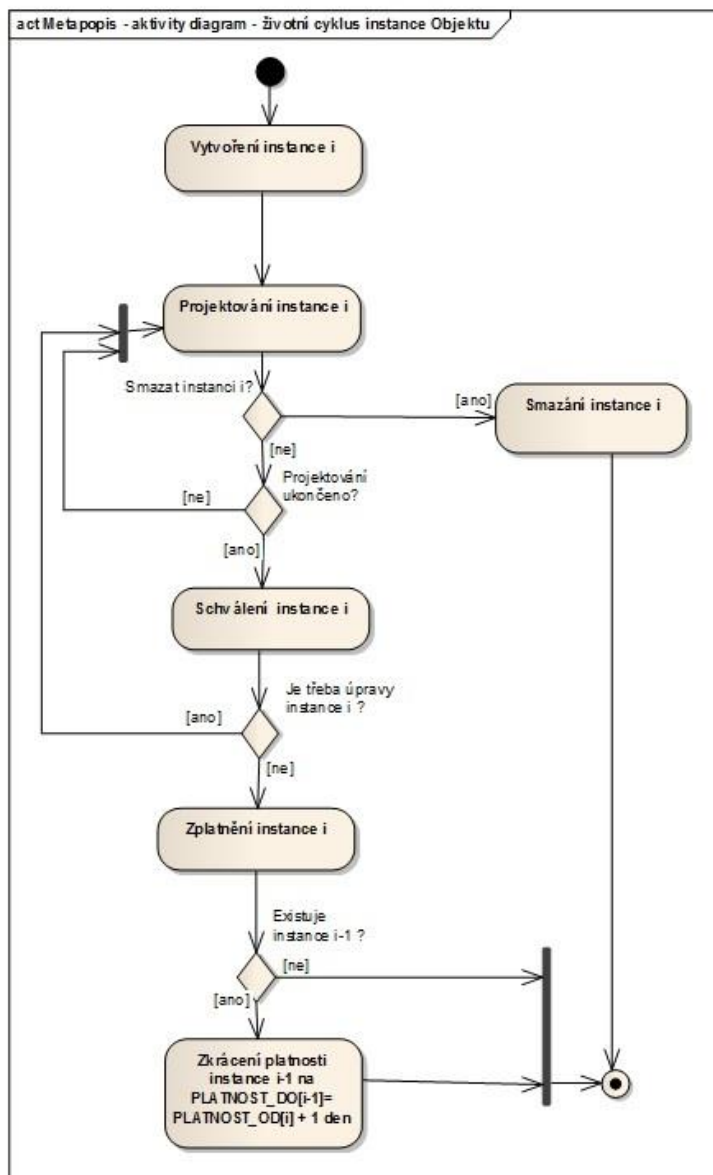
Tabulka 3 - Stavy objektu

#### 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

## 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),
- 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 Přípravný výkazů (viz kapitola [4 Uživatelské rozhraní](#)).

#### **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 Přípravný výkazů a jsou-li prováděny u objektů ve stavu Platný nebo Schválený i z Úložiště schválených a platných výkazů.

#### **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 Přípravný výkazů a ve stanovených případech i z Úložiště schválených a platných výkazů,
- 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 Přípravný výkazů.

#### 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, již byla ukončena platnost tak, aby nebyla narušena časová návaznost verzí nebo variant použitých objektů.

### 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 nadřazené 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 (Pracovní oblasti, Výkazy, Datové oblasti, Úložiště, 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 příloha 4 — Počty instancí vybraných objektů metapopisu v roce 2015 v systému MtS](#).

#### 3.1 Objekt Pracovní oblast

**Sledování historie objektu:** Sledování historie – časová platnost (viz kapitola [2.2.5 Přístup „Sledování historie – časová platnost“](#)).

Účelem objektu Pracovní oblast je seskupit jednotlivé Výkazy do logických celků při jejich tvorbě a následných úpravách. Objekt Pracovní oblast má organizační charakter, tj. slouží primárně k organizaci práce s Výkazy a umožňuje aplikaci hromadných funkcí.

Objekt Pracovní oblast je propojen asociační třídou s objektem Výkaz. Smyslem této vazby je určit, který Výkaz patří do které Pracovní oblasti. Na této vazbě je vymezená časová platnost tohoto zařazení. Platí, že:

- Pracovní oblast může vzniknout, aniž by obsahovala alespoň jeden Výkaz,
- Pracovní oblast může mít přiřazeno N (neomezeně) Výkazů,
- pokud vzniká nový Výkaz nebo je modifikován stávající Výkaz, musí být zařazen (s vymezením platnosti\_od tohoto zařazení) do právě jedné aktivní Pracovní oblasti. Výkaz bez zařazení do Pracovní oblasti nemůže vzniknout,
- pokud je Výkaz zařazován do Pracovní oblasti, je nutno vymežit toto zařazení pomocí určení časové platnosti od – do (pokud uživatel neurčí hodnotu atributu platnost\_do, systém automaticky dosadí 31. 12. 4000),
- pokud je Výkaz zařazován do jiné aktivní Pracovní oblasti, než ve které momentálně je, pak musí dojít k ukončení platnosti přiřazení k původní Pracovní oblasti a vznikne nová definice časové platnosti, která vymezí zařazení Výkazu do nové Pracovní oblasti,
- Pracovní oblast nemůže být smazána, pokud obsahuje jakoukoli (i již ukončenou) vazbu na Výkaz. Aby bylo možno Pracovní oblast smazat, je nutno nejdříve odebrat všechny vazby na objekt Výkaz (vzhledem k tomu, že Výkaz musí vždy být zařazen do alespoň jedné Pracovní oblasti, nelze pouze smazat vazbu mezi Výkazem a Pracovní oblasti, ale je třeba zajistit, aby Výkaz byl rovnou zařazen do jiné Pracovní oblasti nebo smazán).

##### 3.1.1 Atributy objektu Pracovní oblast

Objekt Pracovní oblast obsahuje všechny standardní atributy (viz kapitola [2.4.2 Standardní atributy objektů](#)).

Nad rámec standardních atributů obsahuje objekt Pracovní oblast následující atributy:

- **aktivní (boolean):** vyjadřuje, zda je možné v pracovní oblasti projektovat. Možné hodnoty jsou ANO (lze projektovat), NE (nelze projektovat). Hodnota tohoto atributu je

nastavována systém automaticky na základě pravidel definovaných v procesu definice pracovní oblasti (viz kapitola [5.2 Proces definice Pracovní oblasti](#)).

- **datum zneaktivnění pracovní oblasti (timestamp):** datum a čas, kdy byla zneaktivněna pracovní oblast. Datum a čas přiřazuje systém po schválení nebo smazání nebo převedení do jiné pracovní oblasti všech Výkazů v Pracovní oblasti (tj. v Pracovní oblasti není aktuálně zařazen žádný Výkaz), případně může být funkce zneaktivnění spuštěna uživatelem.
- **autor zneaktivnění (number):** identifikace uživatele nebo SDAT, který zneaktivnil Pracovní oblast. Přiřazuje systém na základě přihlášení do aplikace.

### 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 objektu Výkaz k objektu Pracovní oblast 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,
- Výkaz může mít přiřazeno N (neomezeně) Bloků výkazů,
- pokud vzniká Blok výkazu, musí být zařazen k právě jednomu Výkazu,
- Blok výkazu je vždy přiřazen 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.

Dále platí, že Výkaz má vazbu na objekt Vykazovací rámec, přičemž platí:

- účelem objektu Vykazovací rámec je umožnit jednotlivé Výkazy „označkovat“ tím, že jim bude přiřazen konkrétní Vykazovací rámec, tj. věcná oblast, která je spojuje. Přiřazením Vykazovacího rámce vzniká „označení“ Výkazu a uživatel se pak může doptávat na seznam výkazů zařazených do konkrétního Vykazovacího rámce,
- vazba Výkazu na Vykazovací rámec obsahuje informaci o časové platnosti tohoto zařazení. Výkazu je Vykazovací rámec tedy přiřazován s platností od – do. Pokud již výkaz nemá být nadále zařazen do Vykazovacího rámce, není informace o zařazení smazána, ale je pouze ukončena platnost zařazení,
- objekt Vykazovací rámec plní kromě role spojování Výkazů do stejné věcné oblasti ještě roli, pomocí které lze dynamicky definovat rozsah oprávnění k Výkazům. Aby bylo možno spojit definici rozsahu oprávnění a věcné členění Výkazů do Vykazovacích rámců do jednoho objektu, je v rámci objektu Zařazení výkazu do vykazovacího rámce zaveden atribut „ponechat přístup k datům“, jehož účel je rozebrán v dokumentu [F – Uživatelé a přístupová práva, kapitola 2.8 Objekt Rozsah oprávnění](#), kde je podrobně popsán aparát definice rozsahu oprávnění,
- jeden výkaz může být zařazen do více Vykazovacích rámců, a to i v jeden časový okamžik (překryvy platnosti jednotlivých zařazení jsou povoleny),
- Výkaz smí vzniknout bez přiřazení Vykazovacího rámce,
- jedna instance objektu Vykazovací rámec smí být přiřazena více Výkazům,
- pokud dojde ke smazání Výkazu, dojde k odstranění všech existujících vazeb na Vykazovací rámec, samotný Vykazovací rámec nebude smazáním Výkazu jakkoli modifikován.

### 3.2.1.1 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 Výkaz podléhá kontrole celkové konzistence nebo ne,
- **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.

## 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.

Základem Bloku výkazu je grafická podoba skupiny Datových oblastí zařazených do Bloku výkazu. Grafická struktura je předpokládána ve formátu listu Excelu, případně ve formě Data Point Modelu (dále někde jen DPM) nebo příslušné linkbase XBRL taxonomie.

Řešení pro DPM a taxonomie XBRL je specifikováno v kapitole [6.1 Proces Přebírání metapopisu z externích zdrojů](#).

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

- Blok výkazu může vzniknout, aniž by obsahoval alespoň jednu Datovou oblast,
- Blok výkazu může mít přiřazeno N (neomezeně) Datových oblastí,
- pokud vzniká Datová oblast, musí být zařazena k právě jednomu Bloku výkazu,
- Datová oblast je vždy přiřazena 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),
- Blok výkazu musí vzniknout i tehdy, pokud má být v rámci Výkazu jediný. Bez vzniku tohoto objektu nebude možné do Výkazu zařadit další podřízené objekty (Datová oblast, Údaj).

### **3.3.1.1 Atributy objektu Blok výkazu**

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

- Garant

Nad rámec standardních atributů je u objektu Blok výkazu definován další atribut:

- **pořadí:** pořadí Bloku výkazu v rámci Výkazu.

## **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 dynamický rozměr Datové oblasti a je určena Hierarchií číselníku, Doménou číselníku nebo Doménou datového typu definováním Konkretizovaného parametru nebo 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,



Sektor.státní sféra			Objem hypotečních úvěrů	Objem spotřebitelských úvěrů
A	B	C	1	2
Celkem			1	2

Sektor.banky			Objem hypotečních úvěrů	Objem spotřebitelských úvěrů
A	B	C	1	2
Celkem			1	2

Sektor.obyvateľstvo			Objem hypotečních úvěrů	Objem spotřebitelských úvěrů
A	B	C	1	2
Celkem			1	2
Do splatnosti	Koruna česká	2		
	Euro	3		
	Dolar	4		
	Libra	5		
	Rubl	6		
Po splatnosti	Koruna česká	7		
	Euro	8		
	Dolar	9		
	Libra	10		
	Rubl	11		

Obrázek 10 - Příklad layoutu kartotékové Datové oblasti

System 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 [3.19. 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 konkrétních dimenzionálních Parametrů. Pro každý typ datové oblasti jsou nastavena pravidla projektování podle typu zvolené Datové oblasti,

- **pořadí:** pořadí Datové oblasti v rámci Bloku výkazu.

## 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 možná 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 specifickou vazbu 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“.

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  $\Sigma$  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 nadřazenosti a podřazenosti 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:

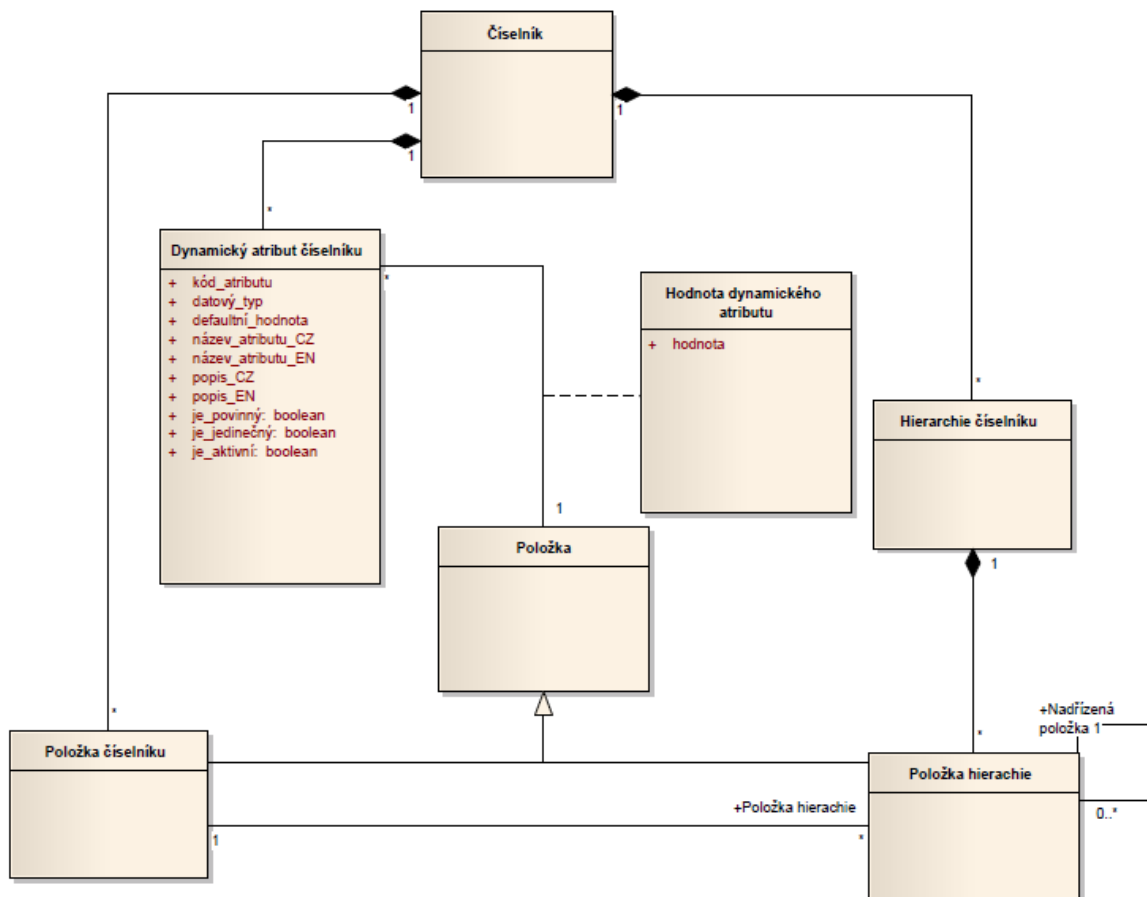


- Datový typ: určuje, jaké hodnoty jsou povoleny pro daný atribut. Jsou k dispozici pouze tyto základní datové typy:
  - NUMBER
  - STRING
  - DATE,
- defaultní hodnota: umožní definovat, jaká hodnota se bude v daném atributu defaultně nastavovat při vytváření nové Položky číselníku. Tento atribut musí 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 (nadřizenost a podřizenost) 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 ( $\Sigma$ ) 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:** nadřizovaná Položka hierarchie, která má další rozpad uvnitř Hierarchie číselníku; uzlová položka je vždy označena znakem ( $\Sigma$ ),



- **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 nadřizenou položku, a to tak, že jako nového rodiče nastaví položku, která je přímo nadřizená 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 ( $\Sigma$ ) a je možné si přímo z Číselníku zobrazit její elementární rozpad,
- v rámci jednoho Číselníku smí být jen jedna Hierarchie číselníku nesynchronizovaná (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 popsány 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“,
- **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 pomoci 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,
- 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.

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 ( $\Sigma$ ) k Položce číselníku (tj. vytvoří součtovou Položku číselníku),
  - v případě změny uzlové položky na elementární položku plošně ve všech instancích Hierarchie číselníku odebere znak ( $\Sigma$ ) 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,
- 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,
  - 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),
  - zobrazí uživateli diagnostiku dopadu na ostatní objekty, které budou změnou dané Hierarchie číselníku dotčeny (Domény číselníku, Datové oblasti, Ukazatele), včetně

- informace, zda budou vytvořeny nové verze či varianty dotčených instancí objektů (Doména číselníku, Datová oblast, Ukazatel),
- v případě, že byla při vytvoření nebo modifikaci Hierarchie číselníku provedena akce, která vede ke konfliktu (např. záměna uzlů, duplicita položek (viz příklady v příloze [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,
    - 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. 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 nadřazených 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 nadřazených 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 vložím 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 ( $\Sigma$ ).
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 nadřizenosti nebo podřizenosti (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 nadřizenosti a podřizenosti (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 „nadřizená/podřizená položka“. Tento vztah určuje, že jedna instance objektu Položka hierarchie je nadřizená jiné instanci objektu Položka hierarchie, tedy že Položka hierarchie „ČR“ má definovanou nadřizenou 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 nadřizená položka):

- každá Položka hierarchie smí mít určenou maximálně jednu Položku hierarchie jako nadřizenou uzlovou položku. Nadřizenou položkou se myslí přímá nadřizená položka (nepřímých nadřizených položek smí mít jedna Položka hierarchie více jak jednu),

- v každé Hierarchii číselníku existuje právě jedna instance třídy Položka hierarchie, která nemá přímou nadřizenou Položku hierarchie. Taková položka se nazývá kořenová položka a v hierarchii položek stojí nejvýše a všechny ostatní Položky hierarchie jsou jí podřízené,
- vazba nadřizená 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:

- **nadřizená položka:** tzv. odvozený atribut, který vznikne na základě nepovinné rekurzivní asociační vazby „nadřizená 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)
<b>H1</b> S_A S_B	<b>H1</b> A1 B1	<b>H1</b> A1 A2 B1 B2	<b>H1</b> S_A A1 A2 B1
<i>Doména je úplná v kořenu H1</i>	<i>Doména je neúplná</i>	<i>Doména je úplná v kořenu H1</i>	<i>Doména je úplná v uzlu S_A</i>

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 nadřazeného 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 objektivě 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 nadřazeného objektu Převodník definován jako „zdrojový číselník“,
- v rámci definice „cílové položky“ mohou být použity pouze ty Položky číselníku z Číselníku, který je v rámci nadřazeného 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 povolena 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 [2.2.6 Přístup „Sledování historie – časová platnost + stavy“](#))

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 „nadřazený 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, pasivní, aktivní i pasivní,
- **nadřazený účet:** odvozený atribut, který vznikne na základě nepovinné rekurzivní asociační vazby „nadřazený 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 = XXXXXXXXXXXXX (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](#).

Objekt Ukazatel se váže na objekt Datová oblast přes asociační třídu Konkretizovaný ukazatel:

- existence instancí obou objektů 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,
- jedna Datová oblast smí obsahovat více Ukazatelů,
- v rámci vazby mezi instancí objektu Datová oblast a instancí objektu Ukazatel vzniká instance objektu Konkretizovaný ukazatel, která dále specifikuje vazbu mezi oběma instancemi, a to takto:
  - jeden Ukazatel smí být do konkrétní Datové oblasti zařazen víckrát, může tak vytvořit více než jednu konkretizaci ukazatele. Jeden Ukazatel smí být zařazen do jedné Datové oblasti víckrát s tím, že může být i stejně konkretizován (instance objektu Konkretizovaný ukazatel). Musí však platit, že každá stejná instance objektu Konkretizovaný ukazatel je jinak dodatečně konkretizována, tzn. je jí přiřazena pokaždé jiná instance objektu Dodatečná konkretizace ukazatele,
  - pokud vzniká instance objektu Konkretizovaný ukazatel, pak:

- musí tato instance mít přiřazen právě jeden z níže uvedených objektů (tím vlastně vzniká ona konkretizace). Účelem je definovat, jaký typ hodnoty v rámci zaslaných dat očekáváme na vstupu (proti této konkretizaci následně také budou postaveny kontroly):
  - Položka (číselníku nebo hierarchie číselníku),
  - Doména (číselníku nebo datového typu),
  - Datový typ,
  - Hierarchie číselníku,
- 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 Konkretizovaný ukazatel tento atribut nastaven na hodnotu „ne“). Pomocí tohoto atributu je definováno, zda k Údaji, který v budoucnu z daného Konkretizovaného ukazatele vznikne, bude povinné zaslat nějakou hodnotu nebo ne.
- Konkretizovaný ukazatel může být zařazen pod jiný, do Datové oblasti již přiřazený, Konkretizovaný ukazatel. Uživatel tak může vytvořit hierarchii Konkretizovaných ukazatelů (o uchování informace o nadřazeném Konkretizovaném ukazateli se stará nepovinná asociační rekurzivní vazba „nadřazený Konkretizovaný ukazatel“), tj.:
  - pokud vzniká hierarchie Konkretizovaných ukazatelů, pak uživatel pro každý Konkretizovaný ukazatel, který obsahuje nějaké podřízené Konkretizované ukazatele, musí definovat, zda výčet podřízených Konkretizovaných ukazatelů je úplný. Tato informace je uchována pomocí atributu „suma“. Na základě hodnoty tohoto atributu jsou pak automaticky generovány kontroly podle následujícího pravidla:
    - pro každý Konkretizovaný ukazatel, který obsahuje nějaké podřízené Konkretizované ukazatele a u kterého je uvedena v atributu „suma“ hodnota „ano“, bude vygenerována kontrola, která bude splněna tehdy, pokud suma hodnot vykázaných na Konkretizovaných ukazatelích, které jsou přímými potomky nadřazeného Konkretizovaného ukazatele, **se rovnají** hodnotě vykázané v nadřazeném Konkretizovaném ukazateli,
    - pro každý Konkretizovaný ukazatel, který obsahuje nějaké podřízené Konkretizované ukazatele a u kterého je uvedena v atributu „suma“ hodnota „ne“, bude vygenerována kontrola, která bude splněna tehdy, pokud suma hodnot vykázaných na Konkretizovaných ukazatelích, které jsou přímými potomky nadřazeného Konkretizovaného ukazatele, **je menší nebo rovna** hodnotě vykázané v nadřazeném Konkretizovaném ukazateli,
- tato instance může (nemusí) být popsána pomocí objektu Účet. Za tím účelem vzniká v objektu Konkretizovaný ukazatel atribut „detailní rozpis

úctů“, jehož účelem je popsat Konkretizovaný ukazatel pomocí odkazu na sadu syntetických účtů. Platí, že:

- při vytváření instance objektu Konkretizovaný ukazatel systém převezme definici vazeb na objekt Účet z objektu Zařazení účtu k ukazateli,
- uživatel může tuto definici upravit nebo zcela odstranit. Atribut „detailní rozpis účtů“ je tak za všech okolností primární z hlediska popisu Konkretizovaného ukazatele. Atribut „detailní rozpis účtů“ je zároveň nepovinný.

### 3.16.1 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:

- **pořadí (number):** obsahuje informaci o tom, v jakém pořadí má být daný Konkretizovaný ukazatel prezentován. Každý Konkretizovaný ukazatel smí mít právě jedno pořadí a toto pořadí musí být jedinečné (nesmí existovat dva Konkretizované ukazatele v jedné Datové oblasti, které by měly stejné pořadí),
- **nadřazený konkretizovaný ukazatel:** odvozený atribut, který vznikne na základě nepovinné asociační rekurzivní vazby „nadřazený Konkretizovaný ukazatel“ a který umožní budovat hierarchii Konkretizovaných ukazatelů,
- **suma (boolean; nemá default): existuje** pouze v případě, že se jedná o Konkretizovaný ukazatel, který má nějaké podřazené Konkretizované ukazatele. Hodnotu zadává uživatel v okamžiku vytváření hierarchických vazeb mezi Konkretizovanými ukazateli a:
  - hodnota „ano“ znamená, že ty Konkretizované ukazatele, které jsou podřazené danému Konkretizovanému ukazateli, představují tzv. „úplný výčet“,
  - hodnota „ne“ znamená, že ty Konkretizované ukazatele, které jsou podřazené danému Konkretizovanému ukazateli, představují tzv. „neúplný výčet“.

### 3.17 Objekt Dodatečná konkretizace ukazatele

Ukazatel, resp. Konkretizovaný ukazatel, je možno rozšířit o další podrobnosti (kromě výše zmíněných). V tu chvíli mluvíme o Dodatečné konkretizaci ukazatele. Dodatečná konkretizace ukazatele umožňuje definici tzv. nespolečných parametrů. Podrobněji je smysl a veškeré podrobnosti tohoto objektu popsán v kapitole [3.19.3 Dodatečná konkretizace ukazatele \(nespolečné Parametry\)](#).

### 3.18 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** konkretizovaných údajů zařazených do Datové oblasti. Existence obou objektů 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,
- 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.

Objekt Parametr se dále váže na objekt Konkretizovaný ukazatel a vytváří tak asociační třídu Dodatečná konkretizace ukazatele. Podrobnosti viz kapitola [3.19.3 Dodatečná konkretizace ukazatele \(nespolečné Parametry\)](#).

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 dokoncretizaci jednotlivých údajů v Datové oblasti.

### 3.18.1 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,
- **pořadí (number):** obsahuje informaci o tom, v jakém pořadí má být daný Konkretizovaný parametr v rámci osy, na kterou je umístěn, prezentován. Každý Konkretizovaný ukazatel 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.

### 3.19 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. 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).

Objekt Údaj v systému může vznikat automatizovaně (generuje ho systém na základě nastavení metapopisu) nebo jej může vytvořit uživatel, 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 Konkretizovaný ukazatel a jedné až N (neomezeně) instance objektu Vykazovaný parametr (viz kapitola [3.19.1 Vznik Údaje – statická Datová oblast](#)),
- může nastat situace, kdy v jiné Datové oblasti vznikne identický Údaj jako již existuje (je popsán stejným Konkretizovaným ukazatelem a stejnou sadou Vykazovaných parametrů). Pokud taková situace nastane, systém toto rozpozná během procesu generování Údajů (viz kapitola [7.15 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.

### 3.19.1 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, 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,
- ukazatel se do Datové oblasti přiřazuje přes objekt **Konkretizovaný ukazatel**. Postup je následující:
  - uživatel určí Datovou oblast, do které chce zařadit Ukazatel,
  - uživatel určí (výběrem ze seznamu) Ukazatel, který chce do Datové oblasti zařadit,
  - během procesu přiřazení Ukazatele do Datové oblasti systém vyžádá od uživatele dodatečnou informaci nutnou k tomu, aby zařazení Ukazatele do Datové oblasti mohlo být uloženo, tzv. konkretizaci ukazatele. Tato konkretizace spočívá v přiřazení právě jedné z následujících informací:
    - Datový typ,
    - Položka (číselníku nebo hierarchie číselníku),
    - Doména (datového typu nebo číselníku),
  - přiřazením této dodatečné informace vzniká instance objektu Konkretizace ukazatele. Pokud použijeme jako Ukazatel „Počet organizačních jednotek vykazující osoby“, pak jej při zařazení do určité Datové oblasti můžeme konkretizovat například datovým typem „N8“ (číslo s maximálním počtem 8 míst, bez desetinné části),
- jeden Parametr smí být do Datové oblasti přiřazen maximálně jednou (a vytvořit tak maximálně jednu konkretizaci Parametru),
- 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. Vyloučeny z možnosti zařadit do Datové oblasti budou i Parametry, které byly přiřazeny přímo k Ukazateli jako tzv. „nespolečné“ (viz další text),
  - 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:
    - 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 přivázan 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 například Doména číselníku, není již možné danému Parametru v rámci dané Datové oblasti přiřadit jinou konkretizaci (tedy Položku číselníku/hierarchie nebo Doménu datového typu/číselníku nebo Hierarchii číselníku). Kombinace Parametru a Datové oblasti 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“,
        - 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, bude doplněno později):
  - Parametr „Měna“/Položka „CZK“,
  - Parametr „Měna“/Položka „EUR“,
  - Parametr „Měna“/Položka „Ostatní bez CZK/EUR“,
  - Parametr „Sektor“/Položka „Rezidenti“,
  - Parametr „Sektor“/Položka „Nerezidenti“,
- necht' 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“,
- každý Údaj je tak popsán právě jedním Konkretizovaným ukazatelem a 1..N (neomezeně, ale alespoň jedním) Vykazovaný 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 Konkretizovaným ukazatelem a právě dvěma Vykazovanými parametry (viz [Obrázek 12 - Konkretizovaný ukazatel a Konkretizovaný parametr](#)),
- celkový počet Údajů ve sledované Datové oblasti je tak 12 (stále zanedbáváme vliv součtových položek hierarchie), protože máme 2 Konkretizované ukazatele a každý z nich vytváří celkem 6 Údajů.

		CZK		EUR		Ostatní bez CZK/EUR	
		Rezidenti	Nerezidenti	Rezidenti	Nerezidenti	Rezidenti	Nerezidenti
A	B	2	3	4	5	6	7
Aktiva celkem	1						
Pasiva celkem	2						

**Údaj UD-1-3:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

---

1. Konkretizovaný parametr: **KP1-1** (Měna.CZK)  
2. Konkretizovaný parametr: **KP2-1** (Sektor.Nerezidenti)

**Údaj UD-1-4:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

---

1. Konkretizovaný parametr: **KP1-1** (Měna.EUR)  
2. Konkretizovaný parametr: **KP2-1** (Sektor.Rezidenti)

Obrázek 12 - Konkretizovaný ukazatel a Konkretizovaný parametr

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,
- 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.

### 3.19.2 Popis Údaje nedimenzionálním parametrem

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

- do Datové oblasti bude zařazen další Parametr, například „Rozsah vykazování“ (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 („ČR“),

- 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  $3 \times 2 \times 1 = 6$  (třetí množina, reprezentovaná parametrem „Rozsah vykazování“ totiž obsahuje jen jeden prvek – číselníkovou položku „ČR“),
- 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 13 - Konkretizovaný ukazatel a Vykazovaný parametr I.](#)).

		Rozsah vykazování ČR						
		CZK		EUR		Ostatní		
		Rezidenti	Nerezidenti	Rezidenti	Nerezidenti	Rezidenti	Nerezidenti	
A	B	1	2	3	4	5	6	7
Aktiva celkem	1							
Pasiva celkem	2							

**Údaj UD-1-3:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

---

1. Vykazovaný parametr: **KP1-1** (Měna.CZK)  
2. Vykazovaný parametr: **KP2-1** (Sektor.Nerezidenti)  
3. Vykazovaný parametr: **KP3-1** (Rozsah vykazování.ČR)

**Údaj UD-1-4:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

---

1. Vykazovaný parametr: **KP1-2** (Měna.EUR)  
2. Vykazovaný parametr: **KP2-2** (Sektor.Rezidenti)  
3. Vykazovaný parametr: **KP3-1** (Rozsah vykazování.ČR)

**Obrázek 13 - Konkretizovaný ukazatel a Vykazovaný parametr I.**

Pokud platí, že Údaj je popsán právě jedním Konkretizovaným 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 popsány a udržují odkaz na stále stejné hodnoty. Prezentační vrstva je tak oddělena od vrstvy popisné a nad jedinou definicí popisu lze vystavět neomezeně různých prezentačních pohledů (červeně je pouze vyznačena změna souřadnic Údaje, nicméně popis je stále tentýž).



		Rozsah vykazování.ČR						
		Nerezidenti			Rezidenti			
		CZK	EUR	Ostatní bez CZK/EUR	CZK	EUR	Ostatní bez CZK/EUR	
A	B	1	2	3	4	5	6	7
Aktiva celkem	1							
Pasiva celkem	2							

**Údaj UD-1-2:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

---

1. Vykazovaný parametr: **KP1-1** (Měna.CZK)  
 2. Vykazovaný parametr: **KP2-1** (Sektor.Nerezidenti)  
 3. Vykazovaný parametr: **KP3-1** (Rozsah vykazování.ČR)

**Údaj UD-1-6:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

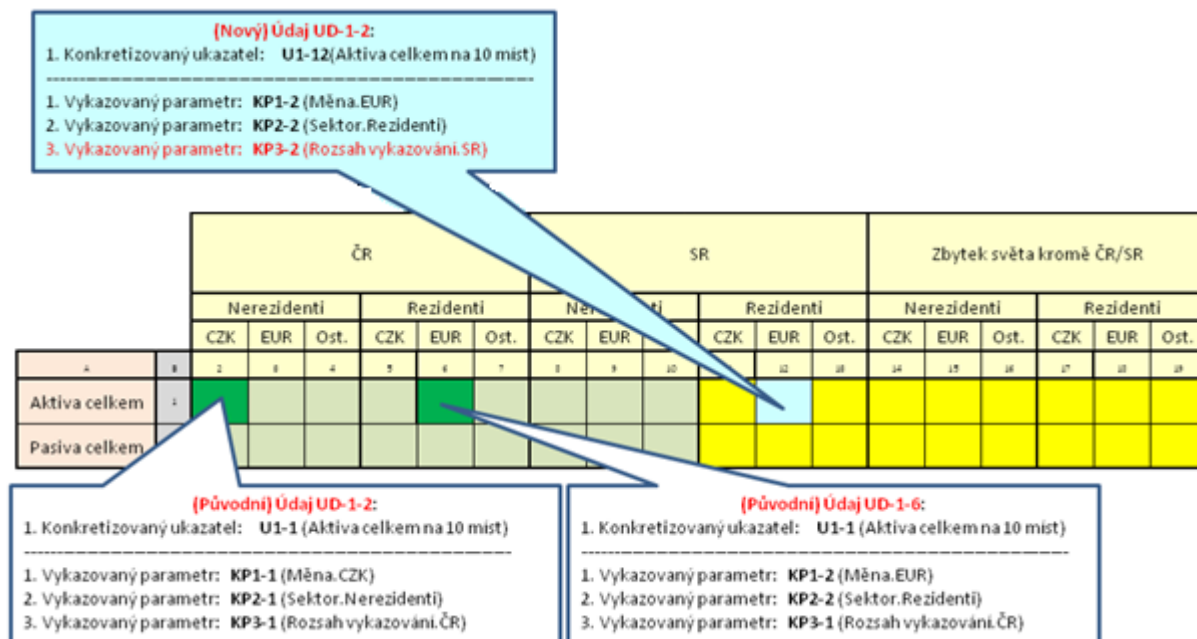
---

1. Vykazovaný parametr: **KP1-1** (Měna.EUR)  
 2. Vykazovaný parametr: **KP2-1** (Sektor.Rezidenti)  
 3. Vykazovaný parametr: **KP3-1** (Rozsah vykazování.ČR)

Obrázek 14 - Konkretizovaný 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 Celý svět/ČR/SR/Ostatní kromě ČR a SR, dojde k výraznému 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 Celý svět/ČR/SR/Ostatní kromě ČR a SR. Protože platí pravidlo o kartézském součinu, pak přestane platit vzorec  $3 \times 2 \times 1 = 6$ , ale tím, že se místo jedné Položky číselníku Parametr P3 konkretizoval pomocí Domény číselníku, která obsahuje 4 položky číselníku, vznikne vzorec  $3 \times 2 \times 3 = 18$  (nenásobíme 4, protože stále vynecháváme sumarizační položku hierarchie „Celý svět“). Ve výsledku tedy k jednomu Konkretizovanému ukazateli vznikne 18 Údajů, celkově jich Datová oblast má 36 (18 Údajů x 2 Konkretizované ukazatele). Protože nová konkretizace Parametru P3 obsahuje i položku „ČR“ (tato položka byla obsažena i v původní konkretizaci), tak se popis původních Údajů nezmění, pouze přibudou Údaje nové (na obrázku 26 zvýrazněné sytou žlutou barvou).





**(Nový) Údaj UD-1-2:**

1. Konkretizovaný ukazatel: **U1-12**(Aktiva celkem na 10 míst)

---

1. Vykazovaný parametr: **KP1-2** (Měna.EUR)
2. Vykazovaný parametr: **KP2-2** (Sektor.Rezidenti)
3. Vykazovaný parametr: **KP3-2** (Rozsah vykazování.SR)

**(Původní) Údaj UD-1-2:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

---

1. Vykazovaný parametr: **KP1-1** (Měna.CZK)
2. Vykazovaný parametr: **KP2-1** (Sektor.Nerezidenti)
3. Vykazovaný parametr: **KP3-1** (Rozsah vykazování.ČR)

**(Původní) Údaj UD-1-6:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

---

1. Vykazovaný parametr: **KP1-2** (Měna.EUR)
2. Vykazovaný parametr: **KP2-2** (Sektor.Rezidenti)
3. Vykazovaný parametr: **KP3-1** (Rozsah vykazování.ČR)

Obrázek 15 - Konkretizovaný ukazatel a Vykazovaný parametr III – nedimenzionální parametr se stává dimenzionálním.

### 3.19.3 Dodatečná konkretizace ukazatele (nespolečné Parametry)

Je třeba také zajistit, aby jeden Konkretizovaný ukazatel mohl být tzv. dodatečně konkretizován. Primárně musí všechny Ukazatele, které jsou zařazeny do Datové oblasti, být konkretizovány (Datovým typem, Doménou datového typu, Doménou číselníku nebo Hierarchií číselníku<sup>2</sup>). Pokud zůstane pouze u této konkretizace, pak bude z daného Konkretizovaného ukazatele vytvořena pouze jedna sada Údajů (bude vytvořeno tolik Údajů, kolik je v dané Datové oblasti Vykazovaných parametrů). Dodatečná konkretizace Ukazatele má zajistit, aby jeden Konkretizovaný ukazatel mohl být konkretizován dalšími (dodatečnými) Parametry, které však budou určeny speciálně pro jeden Konkretizovaný Ukazatel. V modelu je dodatečná konkretizace Konkretizovaného Parametru reprezentována objektem Dodatečná konkretizace ukazatele, přičemž platí:

- každý Konkretizovaný ukazatel smí být dodatečně konkretizován N (neomezeně, tedy i žádným) Parametry. Konkretizované ukazatele tak mohou být dodatečně konkretizovány pomocí různého počtu Parametrů (v našem případě na obrázku 27 je Konkretizovaný ukazatel „Aktiva celkem“ dodatečně konkretizován jedním Parametrem „Splatnost“, který je sám konkretizován Doménou číselníku nad Hierarchií číselníku „Splatnost“, která obsahuje položky Celkem/Krátkodobá/Střednědobá/Dlouhodobá, zatímco Konkretizovaný ukazatel „Pasiva celkem“ není dodatečně konkretizován žádným Parametrem,

<sup>2</sup> V případě konkretizace Ukazatele hierarchickou strukturou (Hierarchie číselníku, Doména číselníku) se abstrahuje od vztahů mezi jejich jednotlivými položkami. Oborem hodnot pro hodnotu ukazatele je tak jednorůvnový seznam položek.

- pro každou Dodatečnou konkretizaci parametru může být použit libovolný Parametr, s výjimkou Parametrů, které jsou použity jako společné (jsou tedy připojeny přímo k Datové oblasti),
- jeden Parametr (pokud není v dané Datové oblasti jako společný) smí dodatečně konkretizovat více Konkretizovaných ukazatelů. Jeden Parametr smí být použit pro Dodatečnou konkretizaci ukazatele maximálně jednou.

V případě, že bude aplikována dodatečná konkretizace Ukazatele, je třeba upravit algoritmus generování Údaje odlišně od předcházejícího případu, ve kterém jsou instance objektu Vykazovaný parametr generovány pouze z třídy Konkretizovaný parametr. Nyní musíme do tohoto objektu uložit i instance, které vzniknou ze třídy Dodatečná konkretizace ukazatele:

- máme Konkretizovaný ukazatel „Aktiva celkem“, který dodatečně konkretizujeme Parametrem „Splatnost“, který je sám konkretizován Doménou číselníku nad Hierarchií číselníku „Splatnost“, která obsahuje čtyři položky Hierarchie číselníku Celkem/Krátkodobá/Střednědobá/Dlouhodobá,
- tato jedna konkretizace Parametru „Splatnost“ musí vytvořit celkem 4 instance třídy Vykazovaný parametr (je to stejný princip jako v případě, kdy se rozpadal objekt Konkretizovaný parametr), **navíc ale je třeba, aby v rámci této instance vznikl odkaz na Konkretizovaný ukazatel, kterého se tato konkretizace týká**. Proto je v modelu vytvořena nepovinná asociační vazba 0..1 mezi objektem Vykazovaný parametr a Konkretizovaný ukazatel,
- zatímco v předcházejícím případě, kdy žádná dodatečná konkretizace neexistovala, se do objektu Konkretizovaný ukazatel vytvořilo pět instancí (bez vazby na jakýkoli Konkretizovaný ukazatel), tedy:
  - Parametr „Měna“/Položka „CZK“,
  - Parametr „Měna“/Položka „EUR“,
  - Parametr „Měna“/Položka „Ostatní bez CZK/EUR“,
  - Parametr „Sektor“/Položka „Nerezidenti“,
  - Parametr „Sektor“/Položka „Rezidenti“,

nyní se k nim musí přidat další tři instance (stále zanedbáváme součtovou položku):

- Parametr „Splatnost“/Položka „Krátkodobá“/Konkretizovaný ukazatel „Aktiva celkem“,
- Parametr „Splatnost“/Položka „Střednědobá“/Konkretizovaný ukazatel „Aktiva celkem“,
- Parametr „Splatnost“/Položka „Dlouhodobá“/Konkretizovaný ukazatel „Aktiva celkem“,

celkově tak objekt Vykazovaný parametr udržuje celkem 8 instancí, z čehož pět jich je společných (nemají odkaz na žádný Konkretizovaný ukazatel) a tři jsou nespolečné (udržují odkaz na právě jeden Konkretizovaný ukazatel),

- musí také dojít k úpravě algoritmu generování Údaje. Zatímco v předcházejícím případě (varianta pouze se společnými Konkretizovanými parametry), byl algoritmus definován:
  - pomocí metody kartézského součinu pronásobit 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 (=výsledná množina) Údaji,

- nově se musí tento algoritmus doplnit o větu: „a zároveň pronásobit všechny prvky výsledné množiny se všemi prvky (Položky) ze všech množin (Parametrů) připojených k jednotlivým Konkretizovaným ukazatelům“,
- jestliže jeden Konkretizovaný ukazatel vygeneroval ve výsledné množině (tvořené pouze společnými Parametry) celkem 6 Údajů a jestliže existuje množina 3 prvků, které dodatečně konkretizují daný Konkretizovaný ukazatel, vznikne v konečném výsledku celkem 18 (6x3 nebo taky 3x3x2) Údajů, které jsou připojeny k danému Konkretizovanému ukazateli. Pokud žádný další Konkretizovaný ukazatel nebude dodatečně konkretizován žádným dalším Parametrem, pak platí, že takový Konkretizovaný ukazatel bude nadále generovat (po zohlednění dodatečné konkretizace) stále 6 Údajů. V rámci prezentace takových dat pak bude dosaženo výsledku na obrázku níže:

		Rozsah Vykazování ČR							
		CZK		EUR		Ostatní bez CZK/EUR			
		Rezidenti	Nerezidenti	Rezidenti	Nerezidenti	Rezidenti	Nerezidenti		
A	Splatnost	B	1	2	3	4	5	6	7
Aktiva celkem	Krátkodobá	1							
	Střednědobá	2							
	Dlouhodobá	3							
Pasiva celkem	N/A	4							

**Údaj UD-1-3:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

---

1. Vykazovaný parametr: **KP1-1** (Měna.CZK)  
 2. Vykazovaný parametr: **KP2-1** (Sektor.Nerezidenti)  
 3. Vykazovaný parametr: **KP3-1** (Rozsah vykazování.ČR)  
 4. Vykazovaný parametr: **KP4-1** (Splatnost.Krátkodobá)

**Údaj UD-3-4:**

1. Konkretizovaný ukazatel: **U1-1** (Aktiva celkem na 10 míst)

---

1. Vykazovaný parametr: **KP1-2** (Měna.EUR)  
 2. Vykazovaný parametr: **KP2-2** (Sektor.Rezidenti)  
 3. Vykazovaný parametr: **KP3-1** (Rozsah vykazování.ČR)  
 4. Vykazovaný parametr: **KP4-3** (Splatnost.Dlouhodobá)

Obrázek 16 - Konkretizovaný ukazatel a Vykazovaný parametr IV – dodatečná konkretizace vybraného Ukazatele

### 3.19.4 Sumarizační (neelementární) položky Hierarchie číselníku

Během dříve uvedeného popisu algoritmu generování Údaje bylo celou dobu záměrně abstrahováno od položek Hierarchie číselníku, které jsou součtové (uzlové). Pokud se tedy budeme držet našeho příkladu, pak v případě Parametru P1 „Měna“, který je konkretizován Doménou číselníku (odvozenou z Hierarchie číselníku) „DOM1“, která obsahuje položky hierarchie:

- Všechny měny,
  - CZK,
  - EUR,
  - Ostatní měny mimo CZK/EUR,

pak je Položka hierarchie „Všechny měny“ tzv. uzlová. Jinými slovy, pokud Vykazující osoba vykazuje data za daný Konkretizovaný ukazatel, pak se součet hodnot uvedený za „CZK/EUR/Ostatní“ musí rovnat hodnotě uvedené ve „Všechny měny“. Toto tvrzení platí za

předpokladu, že Doména číselníku nad Hierarchií číselníku je postavena tak, že obsahuje tzv. „úplný výčet položek“, tedy že nadřizený (sumarizační) uzel hierarchie má v Doméně číselníku zařazený všechny elementární položky nebo je použita celá Hierarchie číselníku.

Existuje požadavek, aby Doména číselníku nad Hierarchií číselníku nebo samotná Hierarchie číselníku mohla mít více uzlů hierarchie.

Z hlediska algoritmu pro generování Údaje mají takové položky speciální postavení:

- systém v první řadě vygeneruje Údaje pro všechny elementární položky (viz příklady výše, kdy jsou z kartézského součinu vyloučeny všechny případné součtové položky),
- systém z každé neelementární Položky hierarchie vytvoří instanci třídy Vykazovaný parametr, ale nastaví mu atribut „suma“ na „ano“. K dříve zmíněným instancím třídy Vykazovaný Parametr tak přibudou další dvě položky:
  - Parametr „Měna“/Položka „Všechny měny“/Suma = ano,
  - Parametr „Sektor“/Položka „Nerezidenti“/Suma = ano,
- tato akce se odehraje ještě před samotným generováním Údaje. Z procesu generování Údaje ale budou výše uvedené (suma = ano) položky vyloučeny, nebudou tak v první fázi tvorby Datové oblasti součástí žádného Údaje,
- po dokončení akce generování Údajů bude moci uživatel dogenerovat jakoukoli další kombinaci uzlové položky Číselníku s elementární Položkou číselníku, případně jakoukoli kombinaci uzlové Položky číselníku s jinou uzlovou Položkou číselníku,
- výsledkem této operace bude Údaj popsáný právě jedním Konkretizovaným ukazatelem a 1..N Vykazovanými parametry.

		Rozsah Vykazování ČR						
		Rezidenti Nerezidenti celkem	Nerezidenti			Rezidenti		
		Všechny měny	CZK	EUR	Ostatní bez CZK/EUR	CZK	EUR	Ostatní bez CZK/EUR
A	B	1	2	3	4	5	6	7
Aktiva celkem	1							
Pasiva celkem	2							

Obrázek 17 - Konkretizovaný ukazatel a vykazovaný parametr V.

### 3.19.5 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 Datový typ 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 nadřizené objekty. Je tedy požadováno, aby existovala „citlivost Datové oblasti“ a „citlivost Výkazu“. Tyto nadřizené 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 nadřizeném Údaji. Podrobně je účel tohoto atributu uveden ve funkčním požadavku UDJ\_1.0.

### 3.20 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. Zasláná 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/KČŘ/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.

### 3.20.1 Atributy objektu Kontrola

Objekt Údaj 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 názvu 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).

### 3.20.2 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.

#### 3.20.2.1 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.

#### 3.20.2.2 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.

### 3.20.3 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.

#### 3.20.3.1 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, 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 [3.20.6 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.

### 3.20.3.2 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.

### 3.20.4 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 systémem SDAT](#)).

#### 3.20.4.1 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 systémem SDAT](#)).

### 3.20.5 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 obdobími. 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](#).

### 3.20.5.1 *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 vykázan, 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.

### 3.20.5.2 *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 systémem SDAT](#).

### 3.20.6 *Odchylka v sémantických kontrolách*

Metapopis stanovuje, v jakých násobcích a s jakou přesností Vykazující osoba zasílá jednotlivé Hodnoty údajů (číselně). 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

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

vstupujících do kontroly) následujícím způsobem:

**Tabulka 5 – Odchylky v závislosti na počtu operandů**

A dále dle vzorce:  $1,0639 * \text{druhá odmocnina (M)}$ , 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):

$$U1 + U2 + U3 = U4 \text{ ODCH } 3000;$$

### Příklad 2

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

$$U11 = U12 + \dots + U31 \text{ ODCH } 0,05;$$

### 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.

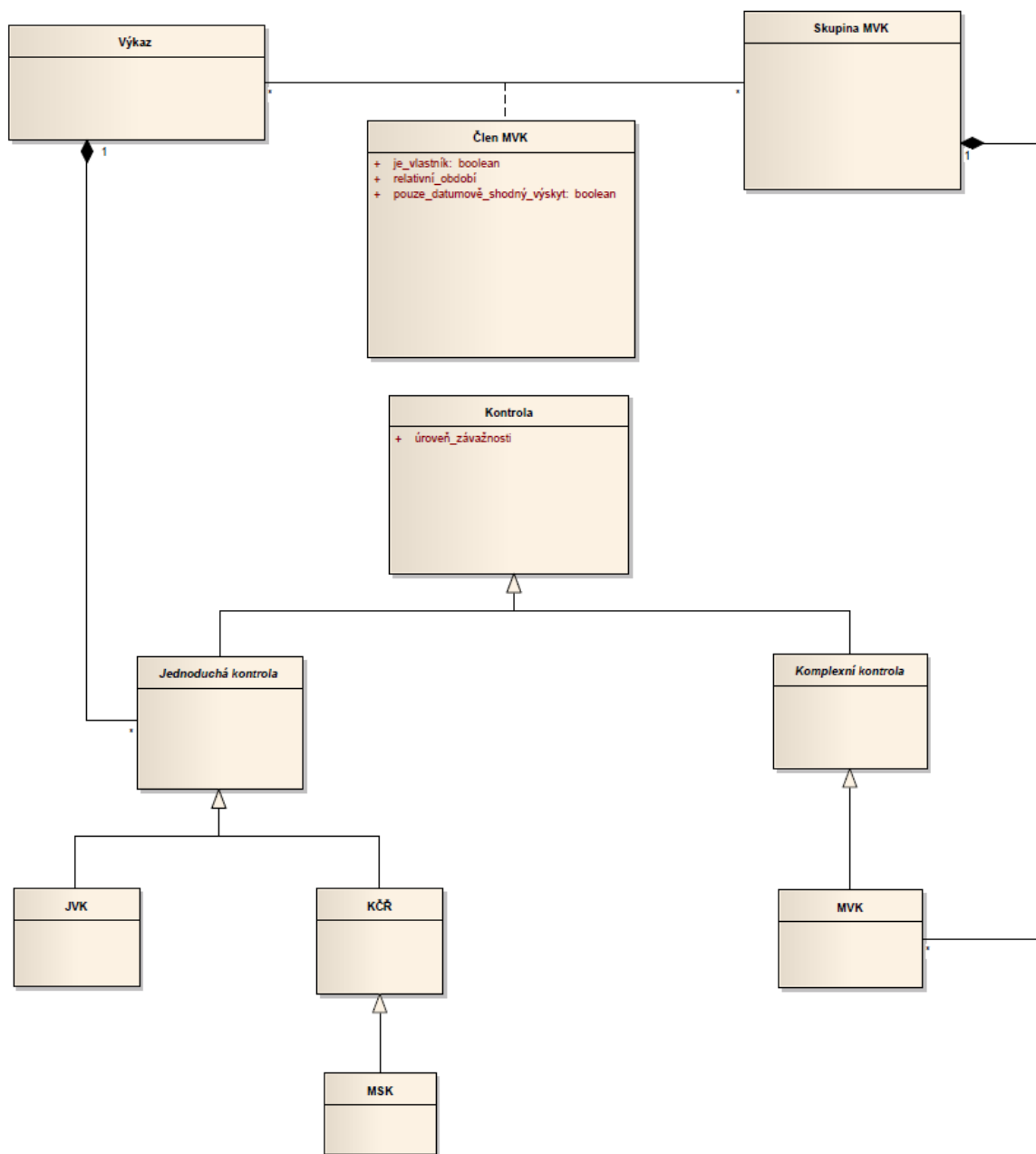
Při generování kontrol z hierarchií systém odchylku stanovuje automaticky pro číselné Hodnoty údajů dle výše uvedeného algoritmu. V případě součtu nad dynamickými Údaji systém generuje odchylku ve výši 10 (+ řád odpovídající násobku vykazovaných hodnot). Uživatel může velikost odchylky upravit podle 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.

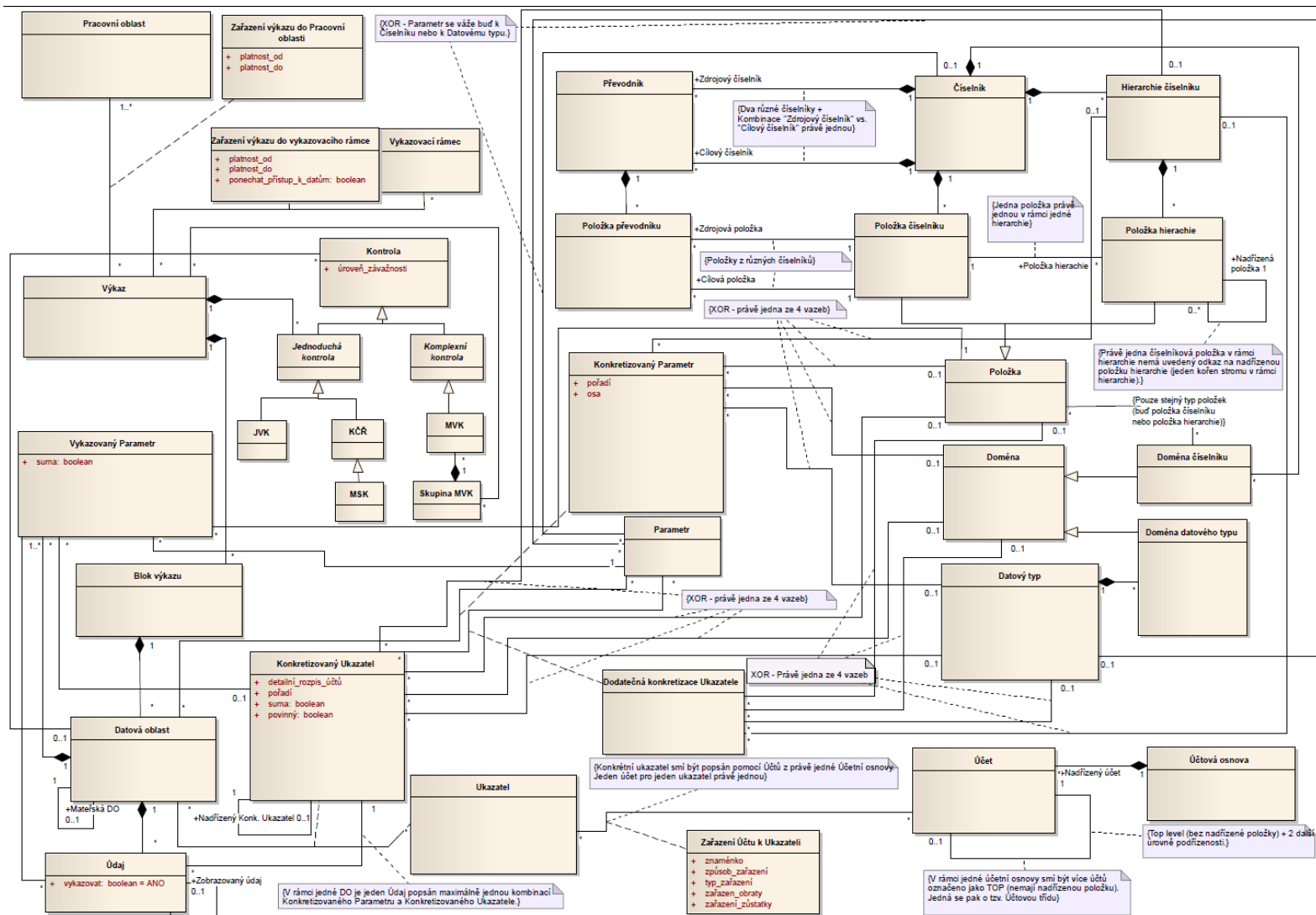
### 3.21 Objektový model – Definice kontrol



Obrázek 18 - Objektový model - Kontroly



### 3.22 Objektový model – Metapopis



Obrázek 19 - Objektový model metapopis

## 4 Uživatelské rozhraní

V návaznosti na pracovní postupy, které je třeba v rámci oblasti metapopisu realizovat, je potřeba navrhnout efektivní uživatelské rozhraní aplikace. Cílem je vytvořit takové uživatelské rozhraní, které oddělí do samostatných okruhů odlišné činnosti, které jsou vykonávány v rámci projektování výkazů a práce s metadaty. Používané procesy z organizačního hlediska vyžadují dále popsané základní okruhy uživatelského rozhraní.

### 4.1 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,
- 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 jiném okruhu uživatelského rozhraní 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).

### 4.2 Přípravná výkazů

Přípravnou výkazů se rozumí okruh uživatelského rozhraní, ve kterém uživatel pracuje na přípravě nových nebo modifikaci stávajících Výkazů, a ukončuje platnost stávajícím Výkazům. Přípravná výkazů dále obsahuje Pracovní oblasti, v rámci kterých jsou jednotlivé

Výkazy nebo jejich skupiny připravovány. V daný okamžik Přípravná výkazů obsahuje 0 až N Pracovních oblastí, které jsou v aktivním stavu.

V rámci práce v okruhu Přípravná výkazů (projektování výkazů) uživatel pracuje s objekty, které jsou primárně spravovány v okruhu Knihovna. Systém umožňuje uživateli používat všechny objekty/instance objektů z okruhu Knihovna přímo v okruhu Přípravná výkazů; v případě potřeby uživatele vytvářet/modifikovat instance objektů z okruhu Knihovna umožňuje systém uživateli přechod do okruhu Knihovna, kde modifikaci provede a návrat zpět do místa, odkud okruh Příprava výkazů opustil.

Pokud v Přípravně výkazů vznikne instance objektu (nebo jeho nová verze/varianta), automaticky přebírá atribut platnost\_od Pracovní oblasti, ze které byl vytvořen, je však umožněno uživateli v případě potřeby toto datum změnit.

### 4.3 Úložiště schválených a platných výkazů

Úložiště schválených a platných výkazů je okruh uživatelského rozhraní, ve kterém jsou dostupné všechny schválené a platné výkazy, které jsou dále distribuovány uživatelům.

V rámci Úložiště schválených a platných výkazů musí být zajištěny následující aktivity:

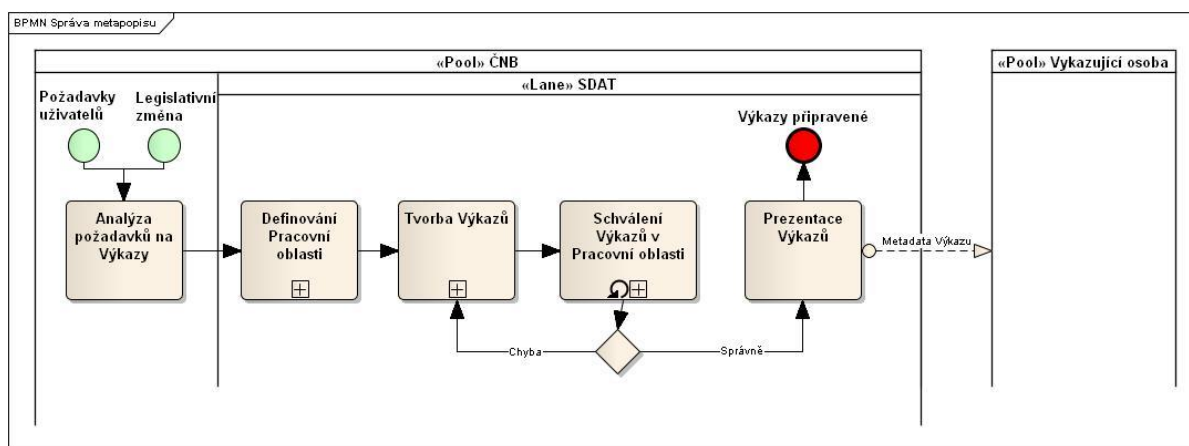
- jsou zde zobrazeny všechny verze schválených a platných Výkazů v celé historii,
- je umožněno prohledávání prostřednictvím nastavených kritérií a jejich kombinací (viz dokument [A – Obecné požadavky, kapitola 3.8.1 Komponenta Tabulka \(grid\)](#)),
- spolu s Výkazy je k dispozici i detailní metodický popis,
- je možné provádět úpravy metainformací, 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 (změnový metapopis). V případě úprav, které mají vliv na obsah údajů, je toto nutno vykonávat v okruhu Přípravná výkazů. Systém umožňuje uživateli přechod do okruhu Přípravná výkazů, kde modifikaci provede a návrat zpět do místa, odkud okruh Úložiště schválených a platných výkazů opustil.
- je možné ukončovat platnost Výkazů,
- je možné prodloužit platnost Výkazů změnou atributu platnost\_do na pozdější datum, pouze v případě, že tato změna nevyžaduje vytvoření nové verze/varianty Výkazu (viz kapitola [2.3 Vazby mezi jednotlivými objekty](#)).

## 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 20 - 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 20 - 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

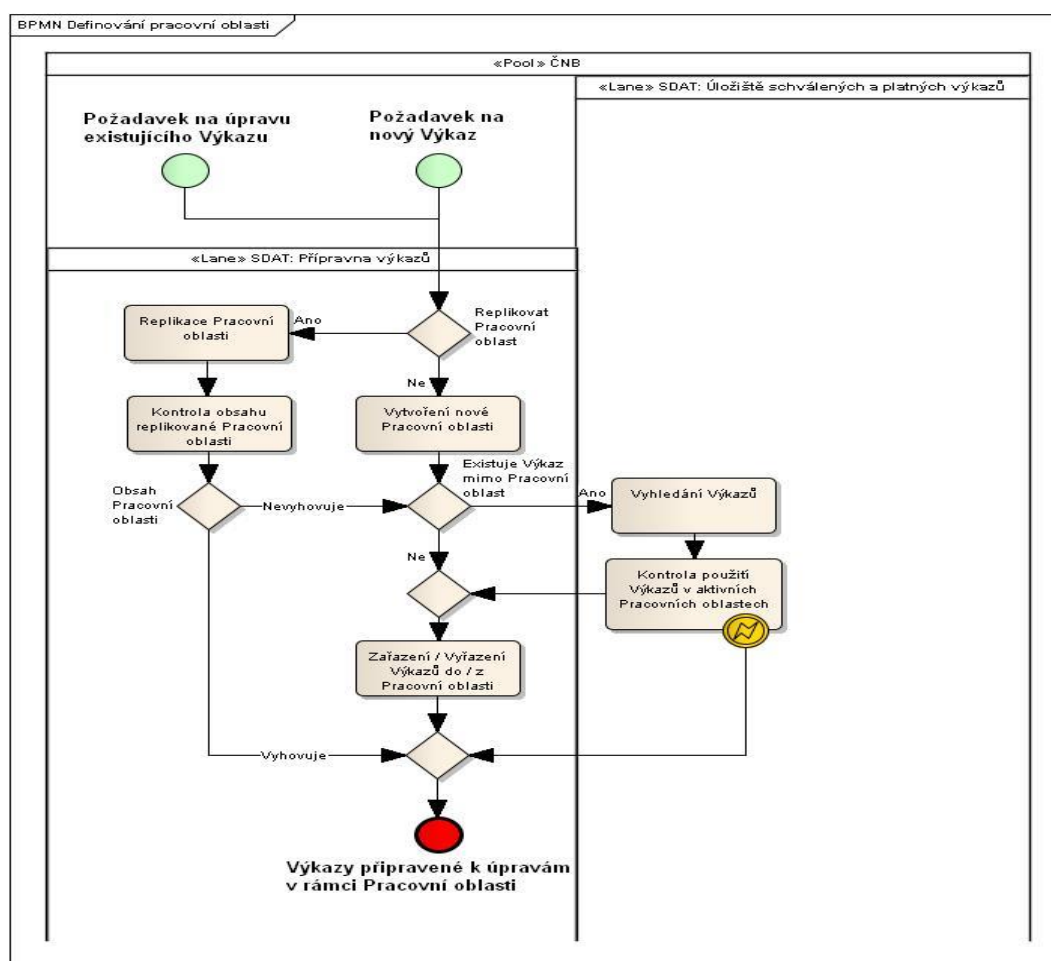
Uživatel vytvoří Pracovní oblast v Přípravně výkazů a naplní ji existujícími platnými Výkazy, které chce změnit nebo jinak upravovat. Kromě toho v ní může zakládat nové Výkazy (viz kapitola [5.2 Proces definice Pracovní oblasti](#)). V rámci této Pracovní oblasti dochází k tzv. tvorbě výkazu, kdy je 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 Pracovní oblasti](#)). 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 Pracovní oblasti

Pracovní oblast (viz kapitola [3.1 Objekt Pracovní oblast](#)) 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 vytvářet Pracovní oblasti a v rámci nich provádět úpravy Výkazů, za které nesou v daném okamžiku zodpovědnost (viz [Obrázek 21 - Proces definice Pracovní oblasti](#)). V jedné Pracovní oblasti 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 2.8.12.1](#) Typ oprávnění „metadata“).



Obrázek 21 - Proces definice Pracovní oblasti

### 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 Přípravně výkazů a je řízen uživatelem. Během rozhodování o potřebě změn Výkazů, které probíhá mimo SDAT, má uživatel možnost si v Úložišti platných a schválených výkazů vyhledávat Výkazy podle zadaných kritérií a zadaná kritéria si podle potřeby ukládat. Při tvorbě Pracovní oblasti lze postupovat dvěma způsoby:

- a) uživatel vytvoří novou Pracovní oblast (viz kapitola [3.1 Objekt Pracovní oblast](#)):
  - uživatel vyplní povinné atributy nové Pracovní oblasti,
  - uživatel naplní vytvořenou Pracovní oblast Výkazy výběrem ze seznamu obsahujícího všechny verze a varianty Výkazů. Filtrování v rámci tohoto seznamu využívá všech možností tabulky (gridu) a navíc obsahuje přepínač pro zobrazení všech, nebo pouze posledních verzí a variant Výkazů,
  - uživatel pro Výkazy zařazované do Pracovní oblasti určí, zda jim má být vytvořena nová verze nebo varianta (viz kapitola [2.3 Vazby mezi jednotlivými objekty](#)); dále musí být splněna podmínka, že Výkaz, který je již zařazen v aktivní Pracovní oblasti, lze zařadit do jiné aktivní Pracovní oblasti pouze pokud je tomuto Výkazu vytvořena další verze nebo varianta, jejíž platnost\_od je vyšší než platnost\_do verze/varianty Výkazu z něhož byla vytvořena, tzn., že takový Výkaz bude mít ve stavu Projektovaný dvě verze/varianty, jejichž datumová platnost se nepřekrývá,
- b) uživatel replikuje Pracovní oblast (viz kapitola [3.1 Objekt Pracovní oblast](#)):
  - uživatel vybere ze seznamu neaktivních Pracovních oblastí požadovanou Pracovní oblast a spustí funkci replikace,
  - uživatel replikované Pracovní oblasti upraví povinné atributy,
  - systém vytvoří repliku původní Pracovní oblasti. Na základě posledních schválených nebo platných verzí/variant Výkazů, které byly obsaženy v původní Pracovní oblasti, nabídne uživateli jejich seznam. Tento seznam neobsahuje Výkazy, kterým byla ukončena platnost, přičemž systém informuje uživatele o Výkazech, jejichž platnost byla ukončena v období od zneaktivnění dané Pracovní oblasti do doby její replikace,
  - uživatel pro Výkazy ze seznamu uvedeného v předchozím bodě určí, zda požaduje vytvoření jejich verze nebo varianty,
  - uživatel zkontroluje, zda vytvořená Pracovní oblast obsahuje všechny jím požadované Výkazy. Pokud v Pracovní oblasti:
    - chybí nějaký Výkaz, uživatel požadovaný Výkaz vyhledá v Úložišti schválených a platných výkazů a zařadí jej do této Pracovní oblasti,
    - je zařazen Výkaz, který by dle rozhodnutí uživatele v této Pracovní oblasti být neměl, uživatel takový Výkaz vyřadí z Pracovní oblasti,
  - uživatel může do replikované Pracovní oblasti zařadit další Výkazy výběrem ze seznamu obsahujícího všechny verze a varianty Výkazů. Filtrování v rámci tohoto



seznamu využívá všech možností tabulky (gridu) a navíc obsahuje přepínač pro zobrazení všech, nebo pouze posledních verzí a variant Výkazů,

- uživatel pro všechny Výkazy zařazované do Pracovní oblasti určí, zda jim má být vytvořena nová verze, nebo varianta (viz kapitola [2.3 Vazby mezi jednotlivými objekty](#)); dále musí být splněna podmínka, že Výkaz, který je již zařazen v aktivní Pracovní oblasti, lze zařadit do jiné aktivní Pracovní oblasti, pouze pokud je tomuto Výkazu vytvořena další verze nebo varianta, jejíž platnost<sub>od</sub> je vyšší než platnost<sub>do</sub> verze/varianty Výkazu z něhož byla vytvořena, tzn., takový Výkaz bude mít ve stavu Projektovaný dvě verze/varianty, jejichž datumová platnost se nepřekrývá,
- uživatel může z replikované Pracovní oblasti vybrané výkazy vyřadit.

Při definování Pracovní oblasti resp. další práce s ní systém:

- kontroluje název zakládáné Pracovní oblasti oproti již existujícím Pracovním oblastem, a to jak aktivním, tak neaktivním, za účelem zabránění tvorby Pracovních oblastí se shodnými názvy; systém neumožní vytvoření Pracovní oblasti se stejným názvem,
- kontroluje název a kód nově vytvářených Výkazů v Pracovní oblasti proti případným tvorbám duplicit; systém neumožní vytvoření Výkazu s již použitým kódem,
- nastaví Pracovní oblasti, po jejím vytvoření, atribut aktivní na ano a zařadí ji do seznamu Pracovních oblastí, který je veden v rámci Přípravný výkazů; seznam Pracovních oblastí je zobrazován ve formě tabulky (gridu), která obsahuje přepínač pro zobrazení všech Pracovních oblastí, nebo pouze aktivních Pracovních oblastí, nebo neaktivních Pracovních oblastí,
- všem verzím nebo variantám Výkazů zařazeným do aktivní Pracovní oblasti 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 přijetí 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 aktivní Pracovní oblasti. To, zda je pro přijetí provedených změn nutné vytvořit verzi nebo variantu Výkazu (viz kapitola [2.3 Vazby mezi jednotlivými objekty](#)) je součástí této informace. Pokud pro přijetí provedených změn:
  - již existuje v aktivní Pracovní oblasti odpovídající verze/varianta Výkazu, Bloku výkazu i Datových oblastí, systém propaguje všechny změny a o provedení této akce informuje uživatele,
  - neexistuje v aktivní Pracovní oblasti odpovídající verze Výkazu, Bloku výkazu a Datových oblastí (např. systém pro propagaci změn vyžaduje verzi Výkazu, Bloku výkazu a Datové oblasti, ale uživatel v Pracovní oblasti 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](#)),
- před zařazením verze/varianty schváleného a platného Výkazu do Pracovní oblasti prověří, zda vybraná verze/varianta Výkazu již není zařazena v jiné aktivní Pracovní oblasti; pokud tuto skutečnost zjistí, oznámí to uživateli s uvedením aktivní Pracovní oblasti, kde je daná verze/varianta Výkazu zařazena, autora této Pracovní oblasti a gestora



věcného obsahu Výkazu; systém nepovolí zařazení stejné verze/varianty Výkazu do dvou aktivních Pracovních oblastí,

- systém vede přehled všech verzí a variant Výkazů, které byly zařazeny do Pracovní oblasti při jejím definování, tj. bez ohledu na to, zda v průběhu existence Pracovní oblasti docházelo k přesunům nebo postupnému schvalování těchto Výkazů. V případě replikované Pracovní oblasti se vede pouze přehled verzí a variant Výkazů v ní ponechaných, nově v ní vytvořených a do ní zařazených, tj. verze/varianty Výkazů, které byly uživatelem vyřazeny z replikované Pracovní oblasti, nejsou součástí tohoto seznamu,
- systém vede aktuální přehled verzí/variant Výkazů zařazených do Pracovní oblasti; tento přehled systém uživateli zobrazí ve formě tabulky (gridu), z něhož je možné otvírat jednotlivé Výkazy, se kterými chce uživatel pracovat,
- po schválení Výkazů v dané Pracovní oblasti nebo jejich smazání či převedení do jiné Pracovní oblasti automaticky danou Pracovní oblast zneaktivní (tj. v Pracovní oblasti nesmí zůstat žádný Výkaz; pokud tam nějaký Výkaz zůstane, musí zůstat Pracovní oblast aktivní). Výkazy lze schvalovat hromadně nebo jednotlivě (viz kapitola [5.6 Proces schválení výkazů v Pracovní oblasti](#)),
- při replikaci Pracovní oblasti vzniká vždy nová Pracovní oblast. Nová Pracovní oblast je systémem označena jako aktivní.

Uživatel může za podpory systému, kdykoliv v průběhu práce s Pracovní oblastí, provádět následující aktivity:

- zařazení verzí/variant Výkazů do Pracovní oblasti:
  - výběrem konkrétní verze/varianty Výkaz ze seznamu obsahujícího všechny verze a varianty Výkazů,
  - vytvořením první verze Výkazů, tj. jejich založením jako objektu (viz kapitola [2.4.5.1 Zakládání instance objektu](#)),
  - přesunutím verze/varianty Výkazu do Pracovní oblasti z jiné aktivní Pracovní oblasti,
- vyřazení verzí/variant Výkazů z Pracovní oblasti výběrem z aktuálního přehledu Výkazů zařazených do Pracovní oblasti,
- 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](#)) v rámci Pracovní oblasti,
- schvalování Výkazů hromadně nebo jednotlivě.

### 5.2.3 Výstup procesu

Je vytvořena aktivní Pracovní oblast obsahující verze/varianty Výkazů, které budou dále upravovány nebo vytvářeny v procesu tvorby výkazů (viz kapitola [5.3 Proces tvorby Výkazu](#)). Tato Pracovní oblast je dostupná uživateli ze seznamu Pracovních oblastí. Uživatel má k dispozici přehled Výkazů v Pracovní oblasti s vybranými atributy, které mu umožní sledovat postup práce na Výkazech.

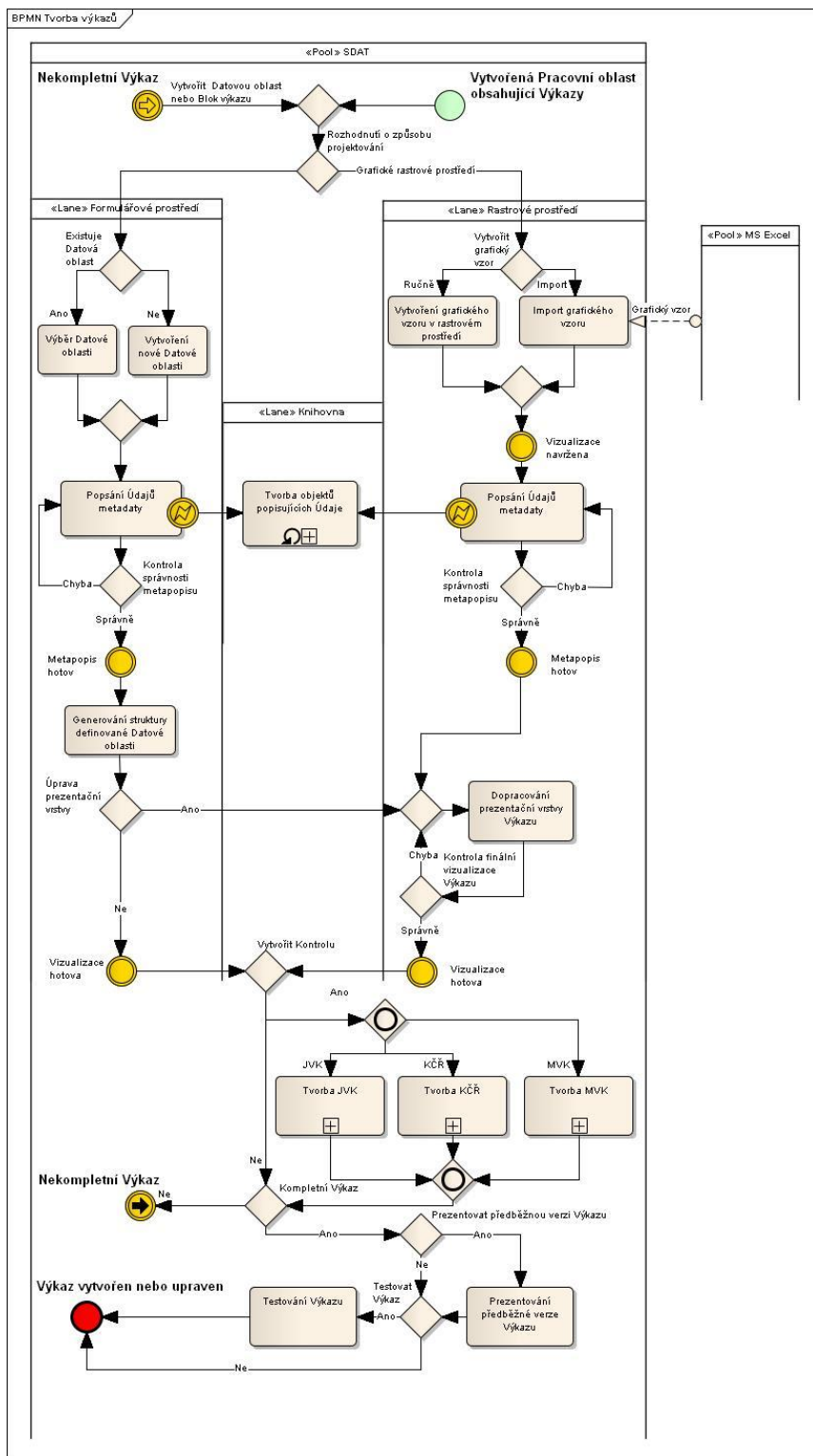
### 5.3 Proces tvorby Výkazu

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 [3.19 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 subprocessy:

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

Z důvodu různé organizace práce je samotná tvorba resp. úprava Výkazů umožněna ve dvou různých prostředích, tj. ve formulářovém prostředí nebo rastrovém prostředí. Výsledkem je však vždy Výkaz a jeho metapopis, který je dále prezentován v rámci systému z věcného hlediska. Pro správné pochopení věcné stránky Výkazu je vytvářena v rámci systému i tzv. prezentační vrstva, která umožňuje dílčím způsobem upravovat grafickou podobu Výkazu resp. jeho Datových oblastí v zájmu jeho srozumitelnosti, tyto úpravy však nesmějí mít vliv na vlastní metapopis. 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 22 - 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 Pracovní oblasti včetně jejího obsahu.

### 5.3.2 Průběh procesu

Proces probíhá v Přípravně výkazů a je řízen uživatelem za podpory systému. Uživatel si vybere prostředí, ve kterém chce vytvářet nový nebo upravovat již existující Výkaz, tj. formulářové (viz kapitola [5.3.2.1 Formulářové prostředí](#)) nebo rastrové prostředí (viz kapitola [5.3.2.2 Rastrové prostředí](#)). Obecně pro obě prostředí platí následující:

- upravovat metapopis lze pouze u Výkazů, které jsou ve stavu Projektovaný a jsou zařazeny do aktivní Pracovní oblasti. Pokud se jedná o úpravu metapopisu Výkazu, jenž byl v minulosti ve stavu Platný, dochází již v okamžiku jeho zařazení do aktivní Pracovní oblasti k 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 má následující možnosti:
  - změny v mateřské Datové oblasti požaduje promítnout do vybraných klonů Datové oblasti (potomků mateřské Datové oblasti): v takovém případě musí uživatel zařadit všechny Výkazy, které obsahují klony Datové oblasti, v kterých požaduje propagovat změnu provedenou v mateřské Datové oblasti, do aktivní Pracovní oblasti. 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](#)); tato tvorba objektů musí být dostupná z obou prostředí pro tvorbu Výkazů v průběhu tvorby výkazu bez jejího přerušení,
- 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,
- systém vyhledává duplicitu v konkretizaci Údajů:
  - v rámci jednoho Výkazu je každý Údaj jedinečný, tj. v případě, že Výkaz obsahuje více Údajů se stejným metapopisem (viz kapitola [3.19 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 Pracovní oblasti,
- 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 z obou prostředí (viz kapitola [5.5 Proces tvorba kontrol výkazu](#)),
- v případě změny Datové oblasti nikdy nedochází, po vygenerování 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 [3.18 Objekt Parametr](#)) v požadovaném pořadí na jednotlivé osy Datové oblasti, příp. na Datovou oblast jako celek, nebo na jednotlivé Údaje individuálně i hromadně na skupinu Údajů. 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,
- vybírat a používat objekty z Knihovny (viz kapitola [4.1 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. 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. 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 musí být z grafického zobrazení Datové oblasti patrné (viz [Obrázek 23 - Zobrazení umístění Ukazatelů a Parametrů v Datové oblasti](#)),

		Ukazatel 1	Ukazatel 2	Ukazatel 3
Parametr 1 (Hodnota 1)	Σ Parametr 2 (Hodnota 1 a Hodnota 2)			
	Parametr 2 (Hodnota 1)			
	Parametr 2 (Hodnota 2)			
Parametr 1 (Hodnota 2)	Σ Parametr 2 (Hodnota 1 a Hodnota 2)			
	Parametr 2 (Hodnota 1)			
	Parametr 2 (Hodnota 2)			

*Hodnota Údaje je popsána Ukazatelem 1, Parametrem 1 (nabývající Hodnotu 1) a Parametrem 2 (nabývající součet Hodnoty 1 a Hodnoty 2)*

*Hodnota Údaje je popsána Ukazatelem 3, Parametrem 1 (nabývající Hodnotu 1) a Parametrem 2 (nabývající Hodnotu 2)*

**Obrázek 23 - 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 není limitováno a umožňuje vzájemnou kombinaci a variabilní pořadí,
- spustit funkci pro vygenerování struktury Datové oblasti, 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. Následně má uživatel možnost vytvořit metapopis pro součtové řádky/sloupce,

- uživatel spustí funkci pro vygenerování Údajů, čímž vzniknou jednotlivé Údaje (viz kapitola [3.19 Objekt Údaj](#)).

V případě, že uživatel u Datové oblasti potřebuje upravit její prezentační vrstvu nad rámec možností struktury Datové oblasti, učiní tak v rastrovém prostředí (viz kapitola [5.3.2.2 Rastrové prostředí](#)).

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

### *5.3.2.2 Rastrové prostředí*

Tvorba Výkazu v rastrovém prostředí je založena na postupném popisu grafického vzoru (tj. grafické vizualizace Výkazu, Bloku výkazu nebo Datové oblasti ve formě tabulky) metadaty, které popisují požadované Údaje. V rastrovém prostředí tedy systém umožňuje:

- vytvořit grafický vzor, který může zahrnovat celý Výkaz (viz kapitola [3.2 Objekt Výkaz](#)) nebo jen jeho konkrétní podřízené části, tj. Blok výkazu (viz kapitola [3.3 Objekt Blok výkazu](#)) nebo Datovou oblast (viz kapitola [3.4 Objekt Datová oblast](#)),
- zobrazit již existující Výkaz včetně podřízených částí; takto zobrazený Výkaz případně jeho část může uživatel dále upravovat, tj. přidávat nebo odebírat sloupce nebo řádky, měnit metapopis jednotlivých Údajů.

Grafický vzor umožňuje víceúrovňové uspořádání Ukazatelů a Parametrů na všech jeho osách ([Obrázek 23 - Zobrazení umístění Ukazatelů a Parametrů v Datové oblasti](#)) a může být:

- vytvořen přímo v systému, nebo
- importován z externího zdroje (viz kapitola [6.1 Proces Přebírání metapopisu z externích zdrojů](#)).

Grafický vzor je možné v systému SDAT dále upravovat a provedené úpravy průběžně ukládat.

Uživatel přímo v rastrovém prostředí v grafickém vzoru označí buňky, ze kterých požaduje vytvoření Ukazatelů a systém:

- hromadně založí požadované Ukazatele (viz kapitola [5.4 Proces tvorby objektů popisujících údaje](#)) v Knihovně, přičemž každému ukazateli předvyplní:
  - kód objektu skládající se z prefixu určeného uživatelem a číselného kódu určeného systémem,
  - název objektu odpovídající textu v označené buňce,
- zbylé atributy vytvořených Ukazatelů má uživatel možnost vyplnit kdykoliv až do okamžiku schválení objektu.

Takto vytvořené Ukazatele jsou přímo umístěny v popisu grafického vzoru na odpovídajících pozicích a uživatel je již nemusí vybírat z Knihovny. Uživatel za účelem dalšího vytvoření metapopisu Údajů přímo v rastrovém prostředí přiřazuje grafickému vzoru příslušné objekty, tj. další Ukazatele, Parametry, které vybírá a používá přímo z Knihovny. Za tímto účelem:

- systém podle zadaných atributů objektů, existujících vazeb objektů, případně jejich kombinace, zobrazí objekty z Knihovny včetně jejich souvisejících vazeb, v časovém kontextu upravovaného Výkazu (platnost\_od),
- systém připojí objekty, které vybral uživatel, ke konkrétním buňkám grafického vzoru,
- uživatel rozdělí grafický vzor do Datových oblastí, a to tak, že označí souvislou skupinu buněk a spustí funkci vytvoření Datové oblasti; po spuštění funkce systém zobrazí formulář Datové oblasti, ve kterém uživatel vyplní minimálně tyto atributy:
  - kód objektu,
  - název,
- zbylé atributy Datové oblasti má uživatel možnost vyplnit kdykoliv až do okamžiku jejího schválení. Pro vyplnění atributů Datové oblasti uživatel vždy musí použít formulář dané Datové oblasti,
- systém vytvoří z buněk grafického vzoru popsanych metadaty Údaje, které zařadí do vytvořených Datových oblastí (viz kapitola [3.4 Objekt Datová oblast](#)), u nichž použití Ukazatelů a Parametrů na jednotlivých osách odpovídá grafickému vzoru (tj. použití Ukazatelů a Parametrů na jednotlivých osách není limitováno a umožňuje vzájemnou kombinaci a variabilní pořadí, přičemž musí být dodrženo, že Ukazatelé mohou být v jedné Datové oblasti pouze na stejné ose),
- pokud metapopis Údajů zařazených do Datových oblastí Výkazu nesplňuje definovaná pravidla metapopisu systému, což je zjištěno kontrolou konzistence, systém uživateli zobrazí zjištěné chyby. Uživatel zjištěné nedostatky v metapopisu musí odstranit, jinak mu systém neumožní verzi/variantu Výkazu schválit.

Vizualizace Výkazu vychází primárně z jednotlivých Datových oblastí nebo Bloků výkazu v případě formulářového prostředí resp. z grafického vzoru v případě rastrového prostředí. Pokud z důvodu srozumitelnosti nebo postižení všech souvislostí z věcného hlediska je výhodné prezentovat Výkaz v odlišné struktuře (např. pokud zjednodušení metapopisu si vyžádalo rozložení dat do několika Datových oblastí, zpřeházení jednotlivých sloupců nebo řádků apod.), je možné provést grafické úpravy, které nemají dopad na samotný metapopis jednotlivých Údajů ani jejich souřadnic určujících pozici Údaje v Datové oblasti. Za tyto grafické úpravy se považuje:

- grafické sloučení dvou nebo více Datových oblastí,
- výměna řádků za sloupce a naopak,
- změna pořadí řádků nebo sloupců,
- doplnění nových řádků a sloupců s pomocnými textovými informacemi (není součástí metapopisu Datové oblasti).

Souřadnice Údajů (viz kapitola [3.19 Objekt Údaj](#)) jsou po provedení těchto úprav nezměněny, tj. souřadnice Údaje odpovídají souřadnicím tohoto Údaje v době, kdy byl v rámci Datové oblasti vyprojektován, viz [Obrázek 24 – Ilustrativní zobrazení souřadnic v Datové oblasti](#). Podle souřadnic Údajů se zobrazuje uživatelský tvar kontroly. Samotné provádění kontrol je závislé na sémantickém tvaru kontroly.

Struktura Datové oblasti stanovena vytvořeným metapopisem Údajů:

	ř/s	Hodnota		
		Brutto	Korekce	Netto
		1	2	3
Aktiva celkem	1			
Nehmotný majetek	2			
Goodwill	3			
Hmotný majetek	4			
Movitý majetek	5			
Nemovitý majetek	6			
Oběžný majetek	7			
Pohledávky	8			
Peníze	9			
Ostatní aktiva	10			

Řádko-sloupcová souřadnice Údaje vyjádřujícího netto hodnotu Nemovitého majetku je [6;3]

Upravená struktura Datové oblasti pro účely prezentace:

	ř/s	Hodnota		
		Netto	Korekce	Brutto
		3	2	1
Aktiva celkem	1			
Nehmotný majetek	2			
Goodwill	3			
Hmotný majetek	4			
Movitý majetek	5			
Nemovitý majetek	6			
Oběžný majetek	7			
Pohledávky	8			
Peníze	9			
Ostatní aktiva	10			

Řádko-sloupcová souřadnice Údaje vyjádřujícího netto hodnotu Nemovitého majetku je [6;3]

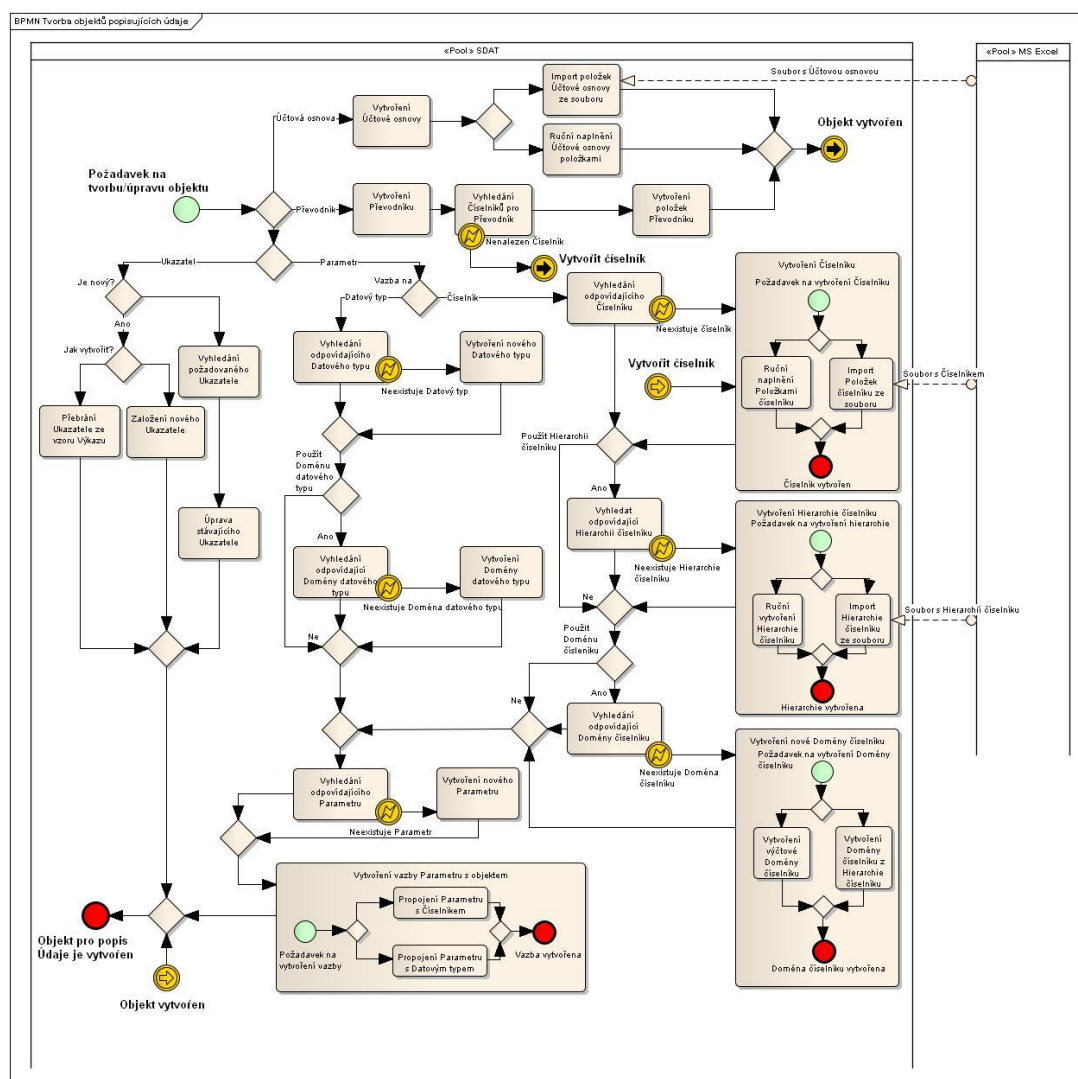
Obrázek 24 – Ilustrativní zobrazení souřadnic v Datové oblasti

### 5.3.3 Výstup procesu

Výstupem je požadovaný Výkaz kompletně popsáný 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 Ukazatelem a Parametry jsou vytvořeny také jim podřízené objekty a nastaveny vazby mezi nimi.



Obrázek 25 - 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 Přebírání 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 Přípravny výkazů, tj. z Pracovní oblasti, kde může uživatel vytvářet objekty jednotlivě nebo hromadně; tvorba objektů je dostupná jak z rastrového, tak i formulářového prostředí. V případě tvorby objektů za pomoci grafického vzoru, která je umožněna v rastrovém prostředí, systém automaticky předvyplní vytvářeným objektům atribut kód objektu a název objektu; tyto atributy je možné dále v systému upravovat.

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 Přebírání 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,
    - uživatel (správce) rozhodne o:
      - potvrzení objektu: tj. potvrdí požadavek a systém změní stav objektu ze stavu Čeká na potvrzení na stav Potvrzený 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 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.



Změna již existujících objektů popisujících Údaje<sup>3</sup> je prováděna uživatelem Změna objektů popisujících Údaje probíhá následovně:

- 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 aktivní Pracovní oblasti, systém informuje uživatele o provedených změnách a zda jejich propagace vyžaduje vytvoření nové verze nebo varianty Výkazu, Bloků výkazu a Datových oblastí. Pokud pro přijetí provedených změn:
      - již existuje v aktivní Pracovní oblasti odpovídající verze/varianta Výkazu, Bloku výkazu i Datových oblastí, systém propaguje všechny změny a o provedení této akce informuje uživatele,
      - neexistuje v aktivní Pracovní oblasti odpovídající verze/varianta Výkazu, Bloku výkazu a Datových oblastí (např. systém pro propagaci změn vyžaduje verzi Výkazu, Bloku výkazu a Datové oblasti, ale uživatel v Pracovní oblasti 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 aktivní Pracovní oblasti, 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 guaranty 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,

---

<sup>3</sup>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.



- v případě, že jsou verze/varianty Výkazů zařazeny do aktivní Pracovní oblasti, systém informuje uživatele o provedených změnách a zda jejich propagace vyžaduje vytvoření nové verze/varianty Výkazu a dotčených Datových oblastí. Pokud pro přijetí provedených změn:
  - již existuje v aktivní Pracovní oblasti odpovídající verze/varianta Výkazu, Bloku výkazu i Datových oblastí, systém propaguje všechny změny a o provedení této akce informuje uživatele,
  - neexistuje v aktivní Pracovní oblasti odpovídající verze Výkazu, Bloku výkazu a Datových oblastí (např. systém pro propagaci změn vyžaduje verzi Výkazu, Bloku výkazu a Datové oblasti, ale uživatel v Pracovní oblasti 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 zařazeny do aktivní Pracovní oblasti, 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 použit 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,
- v případě tvorby objektů z rastrového prostředí je výběr typu objektu umožněn přímo nebo prostřednictvím grafického vzoru, kde je možné označit jednotlivou buňku a určit typ objektu, nebo označit blok buněk a:
  - Ukazatele založit hromadně, tj. více Ukazatelů najednou vč. převzetí některých atributů (např. název),
  - pro vybraný Parametr vytvořit navazující Číselník, Hierarchii číselníku nebo Doménu číselníku z vyznačených položek vč. převzetí některých atributů (např. název); Číselníky, Hierarchie číselníků a Domény číselníků je možné z grafického vzoru vytvářet, pouze pokud jsou v tomto vzoru úplné,
- platnost\_od objektu je defaultně nastavena systémem podle platnost\_od Pracovní oblasti, pokud je z ní objekt vytvářen; uživatel má možnost datum platnost\_od změnit,
- 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,

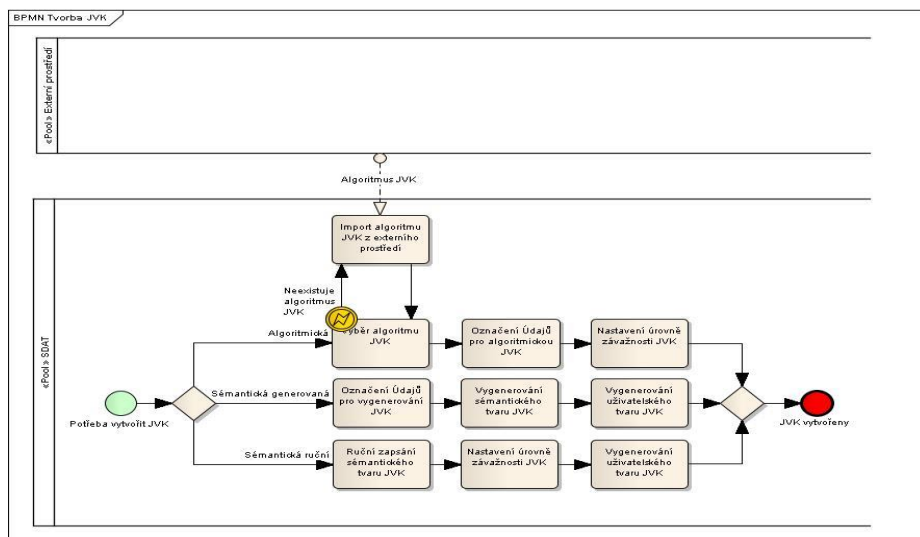
- při výběru z Knihovny se používá výběr ze seznamu (viz dokument [A – Obecné požadavky, kapitola 3.8.1 Komponenta Tabulka dat \(grid\)](#)),
- systém upozorňuje uživatele na vytváření duplicit v Knihovně,
- způsob vyplňování atributů jednotlivých objektů umožňuje:
  - jejich vyplnění dopředu systémem (např. přebírání názvů z grafického vzoru Výkazu, import atributů popis objektu a poznámka z externích zdrojů apod.),
  - úpravu převzatých informací uživatelem,
- Ukazatele se tvoří jednotlivě nebo hromadně, Parametry se tvoří jednotlivě,
- v případě Parametrů je sledována vazba na základní Číselník resp. Položku číselníku nebo Hierarchii číselníku resp. Položku hierarchie, nebo Doménu číselníku, nebo Datový typ resp. Položku datového typu, resp. Doménu datového typu.

### 5.4.3 Výstup procesu

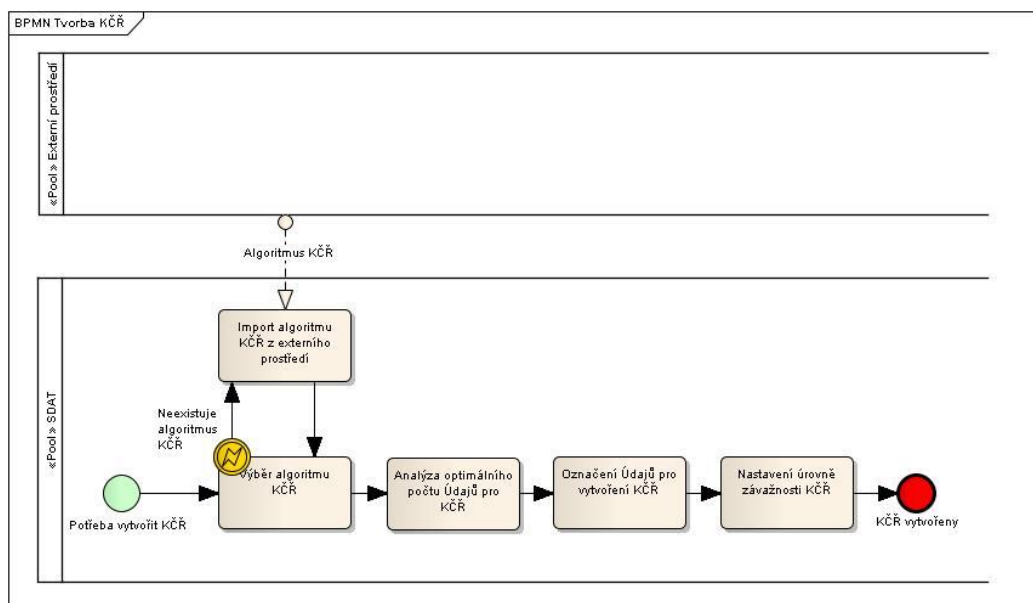
Výstupem procesu je vytvoření všech objektů popisujících Údaje, které jsou potřeba pro tvorbu Výkazů v dané chvíli zařazených v Pracovní oblasti.

### 5.5 Proces tvorba kontrol výkazu

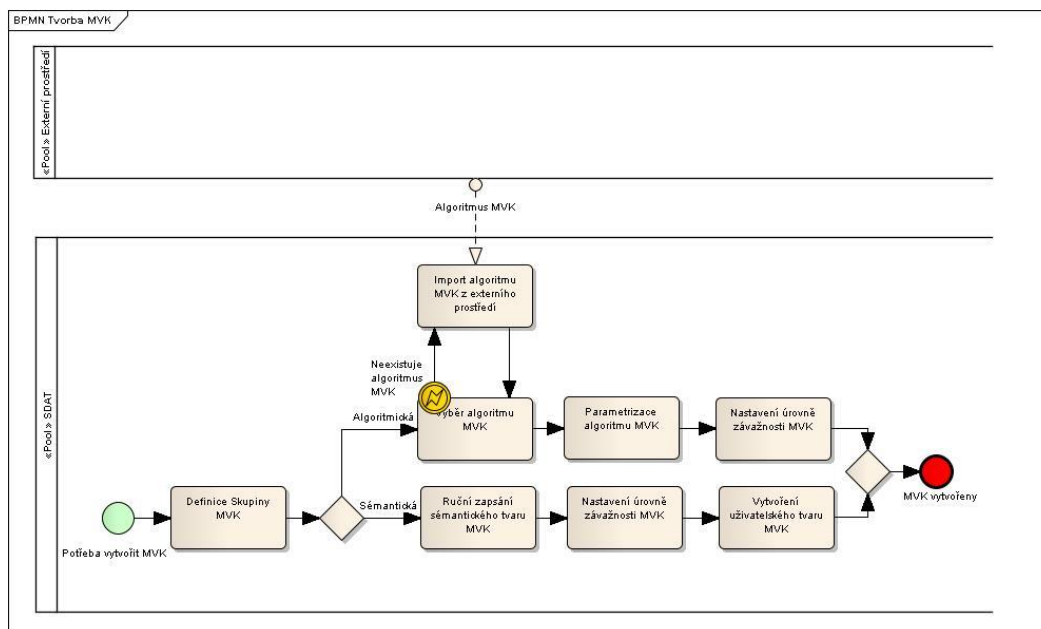
Za účelem zvýšení kvality sbíraných dat je nutné pro vytvořené Výkazy, resp. jejich Datové oblasti definovat kontroly. Tyto kontroly se rozdělují na technické a věcné kontroly. Smyslem těchto kontrol je, aby bylo možné určit, zda přijatá data dosahují požadovanou úroveň kvality a je možné je uložit do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



Obrázek 26 - Proces tvorby JVK



Obrázek 27 - Proces tvorby KČŘ



Obrázek 28 - 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:

- Přípravně výkazů: kontroly jsou pomocí systému tvořeny nebo upravovány pro Výkazy ve stavu Projektovaný,
- Úložišti schválených a platných výkazů: kontroly jsou pomocí systému upravovány pro Výkazy ve stavech Schválený nebo 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 [3.20.6 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 [3.20.1 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 [3.20.3.1 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 [3.20 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 nainportuje, 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 [3.20.1 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 [3.20.1 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:

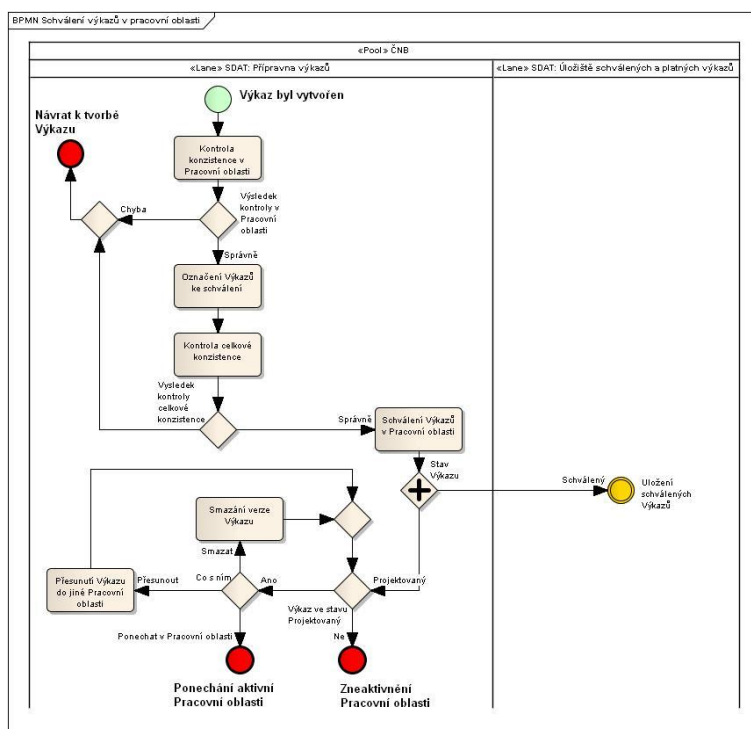
- kontroluje při zadávání kontrol správnost jejich zápisu resp. vyplnění všech potřebných atributů,
- umožňuje automatizované plnění vybraných atributů kontroly,
- podporuje tvorbu názvu kontroly odvozením ze zahrnovaných položek do kontroly podle vybraného algoritmu,
- indikuje kontroly, které musí 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 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 Přebírání 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 Pracovní oblasti

Schválení výkazů v Pracovní oblasti je finální aktivitou tvorby metapopisu. Součástí tohoto procesu jsou kontroly konzistence (viz kapitola [2.5 Kontrola konzistence](#)) Výkazů v Pracovní oblasti, jež předchází vlastnímu schválení.



Obrázek 29 - Proces schválení výkazů v Pracovní oblasti

### 5.6.1 Spouštěč procesu

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

### 5.6.2 Průběh procesu

Proces probíhá v Přípravně výkazů a je řízen uživatelem. Uživatel za podpory systému:

- označí Výkazy určené ke schválení (nebo hromadně označí všechny Výkazy v Pracovní oblasti a odznačí ty, které ještě nechce schválit, tj. to, co je v daném případě podle počtu Výkazů výhodnější) a stanoví pro ně datum platnost\_od, v případě že je požadováno jiné datum platnost\_od než má samotná Pracovní oblast,
- označí Výkazy určené k ukončení platnosti a systém jim nastaví datum platnost\_do na hodnotu data platnost\_od Pracovní oblasti minus jeden den; uživatel má možnost systémem stanovené datum platnost\_do změnit,
- spustí kontrolu konzistence (viz kapitola [2.5 Kontrola konzistence](#)) pro Výkazy vybrané ke schválení v Pracovní oblasti,
- 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 v Pracovní oblasti schválit požadované Výkazy,
  - uživatel neschválí všechny potřebné objekty: v takovém případě nelze v Pracovní oblasti 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 v Pracovní oblasti. Na základě výsledku kontroly konzistence:
  - Výkazy, které neprojdou kontrolou konzistence, nejsou schváleny a jsou zobrazeny uživateli s identifikací zjištěných nedostatků,
  - 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, stav na Schválený a platnost\_od na hodnotu atributu platnost\_od Pracovní oblasti a uloží je do Úložiště schválených a platných výkazů,
  - pokud jsou v Pracovní oblasti schváleny všechny Výkazy, je tato označena jako neaktivní,
- v případě, že v Pracovní oblasti nedošlo ke schválení všech v ní zařazených Výkazů, může nastat následující:
  - Výkazy ve stavu Projektovaný jsou ponechány v Pracovní oblasti, tj. tato Pracovní oblast je ponechána jako aktivní bez časového omezení,
  - Výkazy ve stavu Projektovaný jsou přeřazeny do jiné aktivní Pracovní oblasti, tj. pokud již v této Pracovní oblasti nezůstal žádný Výkaz ve stavu Projektovaný, je Pracovní oblast zneaktivněna,
  - Výkazy ve stavu Projektovaný jsou uživatelem v Pracovní oblasti smazány, tj. pokud již v této Pracovní oblasti nezůstal žádný Výkaz ve stavu Projektovaný, je Pracovní oblast zneaktivněna.

### 5.6.3 Výstup procesu

Výstupem procesu jsou:

- schválené Výkazy, které se nacházejí v Úložišti schválených a platných výkazů a je možné je prezentovat uživatelům,
- neaktivní Pracovní oblast, která uchovává vazby na Výkazy, které v ní byly schváleny.

## 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 odpovědní 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,

		Ukazatel 1	Ukazatel 2	Ukazatel 3
Parametr 1 (Hodnota 1)	Σ Parametr 2 (Hodnota 1 a Hodnota 2)			
	Parametr 2 (Hodnota 1)			
	Parametr 2 (Hodnota 2)			
Parametr 1 (Hodnota 2)	Σ Parametr 2 (Hodnota 1 a Hodnota 2)			
	Parametr 2 (Hodnota 1)			
	Parametr 2 (Hodnota 2)			

Hodnota Údaje je popsána Ukazatelem 1,  
 Parametrem 1 (nabývajícím Hodnotu 1) a Parametrem 2 (nabývajícím součet Hodnoty 1 a Hodnoty 2)

Hodnota Údaje je popsána Ukazatelem 3,  
 Parametrem 1 (nabývajícím Hodnotu 1) a Parametrem 2 (nabývajícím Hodnotu 2)

Obrázek 30 – Ilustrativní zobrazení struktury Datové oblasti

- **prezentační vrstva:** je to vrstva, která vychází ze struktury zobrazovaného výkazu, ale může být dále upravována, aniž by tyto úpravy měly vliv na metapopis Údajů. Pokud uživatel pro účely prezentace výkazu neprovede žádné úpravy, tak prezentační vrstva objektu je vizuálně totožná se strukturou objektu,

Platí pouze pro Vykazující osoby s bilanční sumou nad 30 mil. CZK				
		Ukazatel 1	Ukazatel 2	Ukazatel 3
Parametr 1 (Hodnota 1)	Σ Parametr 2 (Hodnota 1 a Hodnota 2)			
	Parametr 2 (Hodnota 1)			
	Parametr 2 (Hodnota 2)			
Parametr 1 (Hodnota 2)	Σ Parametr 2 (Hodnota 1 a Hodnota 2)			
	Parametr 2 (Hodnota 1)			
	Parametr 2 (Hodnota 2)			

Vložený řádek bez vlivu na metapopis Údajů obsažených v Datové oblasti

Obrázek 31 – Ilustrativní zobrazení prezentační vrstvy Datové oblasti

- 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 Přebírání metapopisu z externích zdrojů

Objekty metapopisu, které jsou přebírány do SDAT z vnějších zdrojů, vznikají na straně externích autorit (např. EBA, EIOPA, ESMA, atd.). Tento metapopis má zpravidla formu Data Point Modelu (DPM) a z něj vytvořené XBRL taxonomie, nebo je dán XML schématem. Obsahuje buď zcela nové objekty anebo objekty s graficky vyznačenými změnami. Metapopis z externího zdroje lze obecně považovat za úplný a není vhodné upravovat jeho věcný obsah. Z toho důvodu je na straně ČNB požadován jeho import. Aby byl tento import do SDAT možný, je nutné, aby všechny importované objekty byly v souladu s principy metapopisu SDAT, čehož uživatel dosáhne úpravou těchto objektů mimo SDAT, případně v průběhu importu externího metapopisu do SDAT. To znamená, že uživatel provede u vybraných objektů změny takovým způsobem, aby tyto objekty odpovídaly pravidlům SDAT pro objekty metapopisu.

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](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](http://www.xbrlwiki.info/index.php?title=European_XBRL_Taxonomy_Architecture_V3.0).

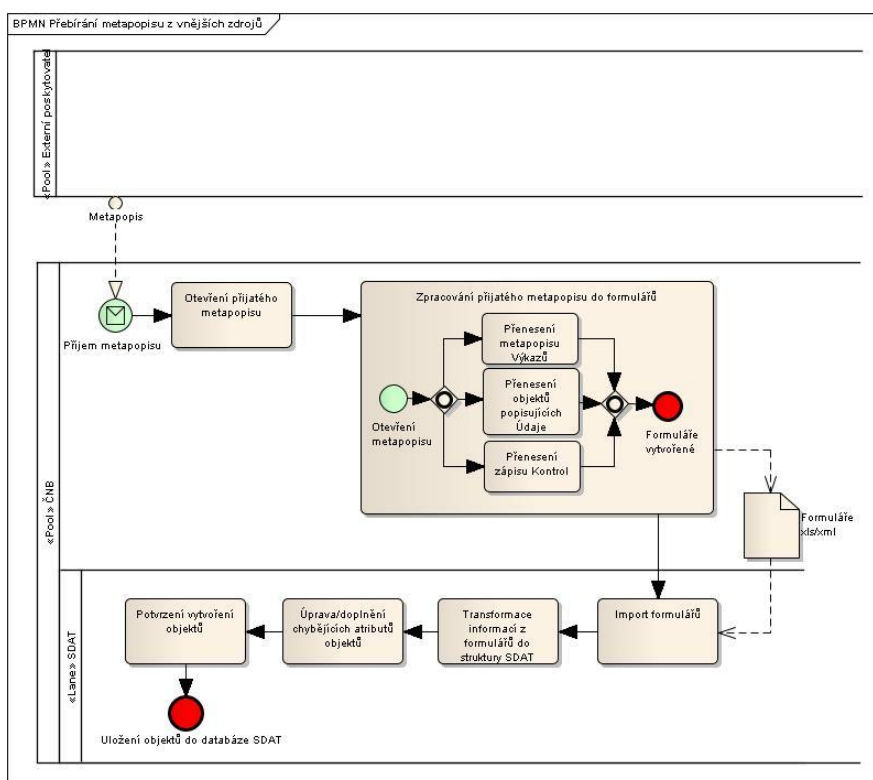
Níže referovaný DPM ve formátu MSAccess (ACCDB), který je v ČNB používán pro přebírání metapopisu z EBA, je tak pouze jednou z implementací konceptuálního modelu DPM. Relační databázové schéma uchovávající metapopis se může měnit. Stejně tak se od sebe mohou lišit technické implementace DPM mezi jednotlivými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

Objekty metapopisu z externího zdroje lze rozdělit do těchto oblastí:

- **grafický vzor Výkazu, nebo Datové oblasti:** metapopis obsahující Výkazy včetně kompletního popisu Údajů metadaty, a to v předem definovaném formátu (XBRL, XML, XLS). Na straně ČNB je nutné tyto Výkazy nebo Datové oblasti vizualizovat a následně přenést jejich grafickou podobu prostřednictvím rastrového prostředí do systému SDAT,
- **objekty určující popis údajů:** metapopis obsahující tzv. slovník, který je součástí DPM nebo taxonomie. Slovník obsahuje Ukazatele, Datové typy, Domény datových typů,

Číselníky, včetně Položek číselníků použitých při popisu Výkazů metadaty. Tento slovník mimo samotné Číselníky obsahuje také Hierarchie číselníků a Domény číselníků,

- **věcné kontroly** definované nad Výkazy, nebo Datovými oblastmi: metapopis obsahující věcné kontroly, které vycházejí z matematických, logických a statistických funkcí aplikovaných v rámci jednotlivých Výkazů nebo Datových oblastí a mezi nimi. Tyto kontroly jsou definovány odkazy na příslušné řádky nebo sloupce Výkazu nebo



Datové oblasti, případně konkrétní Údaje.

Obrázek 32 - Proces Přebírání metapopisu z vnějších zdrojů

### 6.1.1 Spouštěč procesu

Proces je spouštěn uživatelem uvolněním metapopisu na straně externí autority.

### 6.1.2 Průběh procesu

Proces je řízen uživatelem a jeho převážná část probíhá v ČNB mimo systém SDAT. Samotný průběh procesu je následovný:

- uživatel stáhne verzi/variantu metapopisu uvolněnou externí autoritou a uloží ji na určené místo v ČNB,
- následně s použitím vhodného SW nástroje otevře uložený metapopis, tj. vizualizuje jeho obsah,

- uživatel spustí přenos obsahu metapopisu do připravených formulářů, kterých datová struktura odpovídá požadavkům systému SDAT (XLS, XML). Z obdrženého metapopisu může uživatel vytvořit následující formuláře:
  - grafický vzor Výkazu, nebo Datové oblasti,
  - objekty určující popis údajů,
  - věcné kontroly definované nad Výkazy, nebo Datovými oblastmi,
- připravené formuláře obsahující požadované fragmenty externího metapopisu může uživatel dále upravovat, tj. přidávat dodatečné atributy, které jsou potřebné pro import těchto objektů do prostředí SDAT,
- uživatel následně importuje dokončené formuláře do systému SDAT,
- systém na základě importovaných formulářů vytvoří potřebné objekty a zobrazí je uživateli, který provedl import formulářů,
- uživatel může vytvořené objekty v systému SDAT dále upravovat, tj. přidávat případné další atributy jednotlivých objektů (např. popis objektu, pokud nebyl přidán ve vstupním formuláři, atd.),
- následně uživatel potvrdí konečné vytvoření nových objektů na základě importovaných souborů.

### 6.1.3 Výstup procesu

Výstupem procesu je uložení objektů importovaných z externích zdrojů do systému SDAT.

### 6.1.4 Převodník výrazů Data point modelu (DPM)

Převodník výrazů použitých v DPM (příp. XBRL taxonomii) na objekty SDAT zachycuje následující tabulka:

Výraz v DPM	Typ v DPM	Ekvivalent v SDAT
Metric	Primary	Datový typ
Domain	Typed	Datový typ, příp. doména datového typu
Domain	Explicit	Číselník
Domain member	Explicit	Položka číselníku
Dimension		Doména číselníku
Applicable domain		Vazba domény na číselník
Applicable dim		Parametr
Basic concept	Explicit	Ukazatel
Hierarchy		Hierarchie číselníku
Data Point		Údaj
Table		1 – n Datových oblastí

Group of tables		Blok výkazu
Template		Výkaz
Validation/check		Kontrola

**Tabulka 6 – Převodník výrazů DPM na objekty SDAT**

Obdržený DPM se dělí minimálně na dvě části:

- slovník obsahující objekty použité pro popis Údajů: Číselníky, Položky číselníků, Parametry, Domény číselníků (vč. jejich vazeb na konkrétní Číselník), Datové typy, Hierarchie číselníků, Ukazatele,
- grafická podoba Výkazů a Datových oblastí, včetně použití konkrétních objektů ze slovníku. V této části se ke každé buňce DPM váže sada konkrétních objektů popsaných ve slovníku; součástí mohou být taky aplikovatelné věcné kontroly.

Objekty metapopisu jsou tedy do ČNB zasílané ve formátech XBRL (XBRL taxonomie), XLS/ACCDB (Data Point Model), XML. Dosud známé vzory metapopisu vytvářeného externími autoritami jsou k dispozici na následujících odkazech:

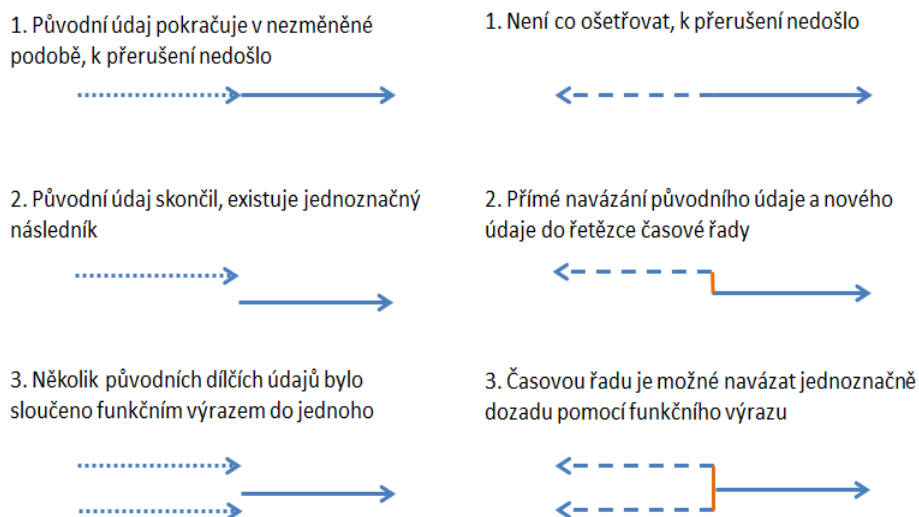
- [https://eiopa.europa.eu/Publications/Reports/EIOPA\\_Solvency\\_2\\_Annotated\\_Templates\\_Sep\\_2012.zip](https://eiopa.europa.eu/Publications/Reports/EIOPA_Solvency_2_Annotated_Templates_Sep_2012.zip),
- <https://www.eba.europa.eu/-/eba-publishes-xbrl-taxonomy-for-remittance-of-supervisory-reporting-by-competent-regulatory-authorities>,
- <http://www.esma.europa.eu/page/Investment-management-0>.

## 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 [3.19 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 33 - 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 uživatel navazuje časovou řadu prostřednictvím funkčního výrazu, vybírá potřebné matematické funkce pomocí průvodce,
- 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 nepřerušovaných 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 časová 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
<b>OBE_1.0</b>	Smazání instance objektu, která podléhá sledování stavů	<p>System umožňuje uživateli smazat jakoukoliv instanci objektu, která je ve stavu Projektovaný (viz kapitola <a href="#">2.4.5.4 Smazání objektů</a>).</p> <p>Smazáním instance objektu jsou automaticky systémem smazány i jemu podřízené instance objektů, které jsou ve stavu Projektovaný.</p> <p>Smazání instance objektu je možné pouze v případě, že daná instance objektu není použita v jiných instancích objektů.</p> <p>Smazání instance objektu je možné pouze v Pracovní oblasti (tj. okruhu Přípravná výkazů), nebo v okruhu Knihovna.</p>	Závazný	1
<b>OBE_2.0</b>	Ukončení platnosti instance objektu uživatelem, která podléhá sledování historie stavů	<p>System umožňuje uživateli ukončit platnost jakékoliv poslední verze/variantě instance objektu, která je ve stavu Platný, změnou atributu platnost_do na požadované datum (viz kapitola <a href="#">2.4.5.3 Ukončování platnosti objektů</a>). Tato akce je povolena pouze v případě, že neexistují další verze/varianty dané instance objektu, které jsou ve stavu Schválený.</p> <p>Ukončení platnosti poslední platné verze/variantě instance objektu je možné pouze v případě, že daná instance objektu není použita v jiných instancích objektů.</p> <p>Ukončením platnosti poslední verze/varianty instance objektu jsou automaticky systémem ukončeny platnosti všech podřízených instancí objektů.</p>	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.		
<b>OBE_2.1</b>	Ukončení platnosti instance objektu systémem, která podléhá sledování historie stavů, v závislosti na zplanění jiné verze/varianty téhož objektu	<p>Systém automaticky ukončí platnost verze/variantě instance objektu, jehož následující verze/variantě byl uživatelem změněn stav na ze stavu Schválený na stav Platný.</p> <p>Systém nastaví atribut platnost_do verze/variantě ukončované instance objektu jako datum platnost_od následné verze/varianty instance objektu minus jeden den.</p> <p>Ukončením platnosti instance objektu jsou automaticky systémem ukončeny platnosti i jemu podřízených instancí objektů, kterým byla zplněna následná verze/varianta.</p> <p>Systém může takto ukončit platnost pouze instancím objektu, které jsou ve stavu Platný a zároveň nejsou použity v jiných instancích objektů.</p> <p>Podrobně je tento způsob zplnění jedné verze/varianty a dopad na existující platnou verzi/variantu popsán v kapitole <a href="#">2.2.6 Přístup „Sledování historie – časová platnost + stavy“</a>.</p>	Závazný	1
<b>OBE_3.0</b>	Prodloužení platnosti instance objektu, která podléhá sledování historie stavů	<p>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 <a href="#">2.4.5.5 Prodloužení platnosti objektu</a>).</p> <p>Prodloužením platnosti instance objektu jsou automaticky systémem prodlouženy platnosti i všem jemu podřízených instancí objektů.</p> <p>Prodloužení platnosti instance objektu je umožněno pouze v případě, že aktuální datum je nižší než datum platnost_do dané instance objektu.</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Prodloužení platnosti instance objektu je možné v okruhu Knihovna a okruhu Úložiště schválených a platných výkazů.		
<b>OBE_4.0</b>	Schválení instance objektu, která podléhá sledování historie stavů	<p>Systém umožňuje uživateli schválit instanci objektu, která je ve stavu Projektovaný, pouze za předpokladu, že kontrola celkové konzistence (viz kapitola <a href="#">2.5 Kontrola konzistence</a>) provedená nad touto instancí objektu skončila bez závažných chyb a případně všechny v ní použité instance objektů knihovny jsou ve stavu Schválený nebo Platný.</p> <p>Schválením instance objektu systém automaticky schválí všechny jemu podřízené instance objektů, které jsou ve stavu Projektovaný.</p> <p>Schválením instance objektu systém změní stav schvalované instance objektu ze stavu Projektovaný na stav Schválený.</p>	Závazný	1
<b>OBE_5.0</b>	Zplatnění instance objektu, která podléhá sledování historie stavů	<p>Systém umožňuje uživateli zplatnit instanci objektu, která je ve stavu Schválený, pouze za předpokladu, že kontrola celkové konzistence (viz kapitola <a href="#">2.5 Kontrola konzistence</a>) provedená nad touto instancí objektu skončila bez závažných chyb a všechny v ní použité instance objektů knihovny jsou ve stavu Platný.</p> <p>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).</p> <p>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.</p>	Závazný	1
<b>OBE_6.0</b>	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)	<p>Systém zároveň změní stav všech jeho podřízených instancí objektů, které jsou ve stavu Schválený na stav Projektovaný.</p> <p>Změna stavu instance objektu ze stavu Schválený na stav Projektovaný je možná v okruhu Knihovna nebo v Pracovní oblasti (tj. v okruhu Přípravná výkazů).</p> <p>V případě, že změna stavu je proveditelná pouze v aktivní Pracovní oblasti a instance objektu má vazbu na neaktivní Pracovní oblast, systém tuto Pracovní oblast zaktivní.</p>		
<b>OBE_7.0</b>	Změna varianty na verzi	<p>Systém umožňuje uživateli změnit vytvořenou variantu instance na verzi instance objektu (viz kapitola <a href="#">2.2.1 Číslo verze a varianty instance objektu</a>).</p> <p>Změna varianty instance objektu na verzi této instance objektu je možná, pouze pokud varianta instance objektu je ve stavu Projektovaný.</p>	Závazný	1
<b>OBE_8.0</b>	Nastavení atributu garant	<p>Systém defaultně do atributu garant instance objektu vyplní identifikaci uživatele, který instanci objektu vytvořil.</p> <p>Systém umožňuje uživateli změnit defaultní nastavení výběrem jiného garanta ze seznamu zaměstnanců ČNB.</p>	Závazný	1
<b>OBE_9.0</b>	Vytvoření nové verze/varianty objektu Projektovaný (u instancí, které podléhají sledování historie stavů)	<p>Systém umožňuje vytvořit instanci (verzi/variantu) objektu za dodržení pravidel v kapitole <a href="#">2.2.1 Číslo verze a varianty instance objektu</a>.</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>OBE_10.0</b>	Smazání instance objektu, která podléhá sledování časové platnosti	<p>Systém umožňuje uživateli smazat jakoukoliv instanci objektu, která nemá vazbu na žádnou instanci objektu, jenž je ve stavu Schválený nebo Platný (viz kapitola <a href="#">2.4.5.4 Smazání objektů</a>).</p> <p>Smazání instance objektu je možné pouze v Pracovní oblasti (tj. okruhu Přípravná výkazů).</p>	Závazný	1
<b>OBE_11.0</b>	Ukončení platnosti instance objektu uživatelem, která podléhá sledování časové platnosti	<p>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 <a href="#">2.4.5.3 Ukončování platnosti objektů</a>). 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.</p> <p>Zároveň systém smaže všechny následující verze/varianty ukončované instance objektu, které mají vazbu na instance objektů, jež jsou ve stavu Projektovaný.</p>	Závazný	1
<b>OBE_11.1</b>	Ukončení platnosti instance objektu systémem, která podléhá sledování časové platnosti, v závislosti na zplanění jiné verze/varianty téhož objektu	<p>Systém automaticky ukončí platnost verze/variantě instance objektu, jehož následující verze/varianta má vazbu na instanci objektu, které byl uživatelem změněn stav na ze stavu Schválený na stav Platný.</p> <p>Systém nastaví atribut platnost_do verze/variantě ukončované instance objektu jako datum platnost_od následné verze/varianty instance objektu mínus jeden den.</p> <p>Systém může takto ukončit platnost pouze instancím objektu, které mají vazbu na instance objektů, jež jsou ve stavu Platný.</p>	Závazný	1
<b>OBE_12.0</b>	Změna varianty na verzi instance	Systém umožňuje uživateli změnit vytvořenou variantu instance na verzi instance objektu (viz kapitola <a href="#">2.2.1 Číslo verze a varianty instance</a>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	objektu, která podléhá sledování časové platnosti	<a href="#">objektu</a> ). Změna varianty instance objektu na verzi této instance objektu je možná, pouze pokud varianta instance objektu má vazbu na instanci objektu, jenž je ve stavu Projektovaný.		

## 7.2 Pracovní oblast

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>POB_1.0</b>	Pracovní oblast - nová	Systém umožňuje uživateli založit instanci objektu Pracovní oblast (viz kapitola <a href="#">5.2 Proces definice Pracovní oblasti bod a</a> ), která je popsána standardními atributy (viz kapitola <a href="#">3.1.1 Atributy objektu Pracovní oblast</a> ) a neobsahuje vazby na žádné verze / varianty Výkazy.	Závazný	1
<b>POB_1.1</b>	Pracovní oblast - replikace	Systém umožňuje uživateli založit instanci objektu Pracovní oblast jako replikaci již existující instance objektu Pracovní oblasti (viz kapitola <a href="#">5.2 Proces definice Pracovní oblasti bod b</a> ).	Závazný	1
<b>POB_1.2</b>	Pracovní oblast – sledování historie	Systém u každé instance objektu Pracovní oblast sleduje její historii podle kapitoly <a href="#">3.1 Objekt Pracovní oblast</a> .	Závazný	1
<b>POB_2.0</b>	Pracovní oblast – atributy nastavené systémem	Systém nastavuje instanci objektu Pracovní oblast vytvořené podle POB_1.0 nebo POB_1.1 následující atributy: <ul style="list-style-type: none"> <li>interní identifikátor objektu,</li> <li>autor objektu (přihlášený uživatel),</li> <li>datum vytvoření (aktuální datum),</li> <li>kdo aktualizoval (přihlášený uživatel),</li> <li>platnost_do (maximální datum),</li> </ul>	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>• garant (viz OBE_8.0),</li> <li>• aktivní (ano).</li> </ul> <p>Systém nastavuje instanci objektu Pracovní oblasti zneaktivněné podle POB_4.0 následující atributy:</p> <ul style="list-style-type: none"> <li>• aktivní (ne),</li> <li>• datum zneaktivnění pracovní oblasti (aktuální datum),</li> <li>• autor zneaktivnění (přihlášený uživatel).</li> </ul>		
<b>POB_2.1</b>	Pracovní oblast – atributy zadávané uživatelem	<p>Systém umožňuje uživateli vyplnit atributy:</p> <ul style="list-style-type: none"> <li>• kód objektu,</li> <li>• název objektu,</li> <li>• popis objektu,</li> <li>• poznámka</li> <li>• platnost_od (viz POB_2.5).</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit.</p>	Závazný	1
<b>POB_2.2</b>	Pracovní oblast – atributy (zadané systémem) měněné uživatelem	<p>Systém umožňuje uživateli měnit následující atributy vyplněné podle POB_2.0:</p> <ul style="list-style-type: none"> <li>• platnost_do (viz POB_2.5),</li> <li>• garant (výběrem ze seznamu zaměstnanců ČNB).</li> </ul>	Závazný	1
<b>POB_2.3</b>	Pracovní oblast – atributy měněné systémem	<p>Systém nastavuje v případě změny některého atributu uživatelem podle POB_2.1 a POB_2.2 nebo přiřazení nové verze / varianty Výkazu podle POB_3.1 a POB_3.2 nové hodnoty atributům:</p> <ul style="list-style-type: none"> <li>• kdo aktualizoval (uživatel, který objekt vytvořil),</li> <li>• datum a čas aktualizace (aktuální datum).</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>POB_2.4</b>	Pracovní oblast – unikátnost atributů	Systém zajišťuje unikátnost atributů kód objektu a název objektu v rámci všech instancí objektu Pracovní oblast. Nepovolí uživateli založit instanci objektu Pracovní oblast s kódem objektu nebo názvem objektu, který je již použit pro jinou instanci objektu Pracovní oblast. Nepovolí uživateli změnit kód objektu nebo název objektu u existující instance objektu Pracovní oblast na hodnotu, která je použita pro jinou instanci objektu Pracovní oblast.	Závazný	1
<b>POB_2.5</b>	Pracovní oblast – vymezení platnosti a dopad na výkaz	V případě, že uživatel změní atributy platnost_od (viz POB_2.1) resp. platnost_do (viz POB_2.2) instance objektu Pracovní oblast, systém tuto změnu promítne do atributů platnost_od resp. platnost_do v ní obsažených instancí objektů Výkaz, které jsou ve stavu Projektovaný tak, aby vymezení platnosti Výkazů (platnost_od až platnost_do) byla v intervalu vymezení platnosti instance objektu Pracovní oblast (platnost_od až platnost_do).	Závazný	1
<b>POB_3.0</b>	Pracovní oblast – vytvoření Výkazu	Systém umožňuje uživateli vytvořit v Pracovní oblasti první verzi Výkazu.  Systém zobrazuje uživateli seznam všech existujících verzí/variant výkazu (viz kapitola <a href="#">5.2 Proces definice Pracovní oblasti</a> ) a umožňuje uživateli vytvořit v Pracovní oblasti verzi/variantu N+1 vybraného Výkazu, kde N, je číslo poslední verze/varianty tohoto Výkazu za dodržení pravidel stanovených v kapitolách <a href="#">2.2.1 Číslo verze a varianty instance objektu</a> a <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .	Závazný	1
<b>POB_3.1</b>	Pracovní oblast – přesun Výkazu	Systém umožňuje uživateli přiřadit k aktivní Pracovní oblasti verzi/variantu Výkazu, která je zařazena v jiné aktivní Pracovní oblasti.	Závazný	1
<b>POB_3.2</b>	Pracovní oblast – počet Výkazů	Systém umožňuje uživateli vytvořit v aktivní Pracovní oblasti neomezené množství verzí/variant různých Výkazů.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
POB_3.3	Pracovní oblast – začátek platnosti vazby na Výkaz	Systém nastavuje začátek platnosti vazby Pracovní oblasti na Výkaz, která vzniká na základě POB_3.0 nebo POB 3.1 na aktuální datum. Tuto hodnotu umožňuje systém uživateli změnit.	Závazný	1
POB_3.4	Pracovní oblast – konec platnosti vazby na Výkaz	Systém nastavuje konec platnosti vazby Pracovní oblasti na Výkaz, která vzniká na základě POB_3.0 nebo POB 3.1 na maximální datum. Tuto hodnotu umožňuje systém uživateli změnit.  Systém nastavuje konec platnosti vazby Pracovní oblasti na Výkaz, která zaniká na základě POB 3.1 na aktuální datum -1. Tuto hodnotu umožňuje systém uživateli změnit.	Závazný	1
POB_3.5	Pracovní oblast – nepřekrývání časové platnosti vazby na Výkaz	Systém zajišťuje nepřekrývání časových platností vazeb verze/variant jednoho Výkazu na různé Pracovní oblasti.	Závazný	1
POB_4.0	Pracovní oblast - zneaktivnění	Systém zneaktivňuje Pracovní oblast v případě, že v důsledku schválení výkazů v Pracovní oblasti (viz VYK_4.2) nezůstane žádná instance objektu Výkaz ve stavu Projektovaný s vazbou na tuto Pracovní oblast.  Systém zneaktivňuje Pracovní oblast v případě, že v důsledku smazání verze/varianty výkazu v Pracovní oblast (viz VYK_6.0) nebo ukončení platnosti vazby verze/varianty výkazu na Pracovní oblast (POB_3.1 a POB 3.4) v Pracovní oblasti nezůstane žádná instance objektu Výkaz ve stavu Projektovaný s vazbou na tuto Pracovní oblast a zároveň existuje aspoň jedna instance objektu Výkaz ve stavu Schválený nebo Platný s vazbou na tuto Pracovní oblast.	Závazný	1
POB_5.0	Pracovní oblast - smazání	Systém umožňuje uživateli smazat instanci objektu Pracovní oblast v případě, že neobsahuje žádnou vazbu na Výkaz bez ohledu na časovou platnost této vazby.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>POB_6.0</b>	Pracovní oblast – zobrazení seznamu	Systém zpřístupňuje Pracovní oblasti v okruhu Přípravná výkazů (viz kapitola <a href="#">4.2 Přípravná výkazů</a> ). Zobrazuje uživateli seznam Pracovních oblastí (viz kapitola <a href="#">5.2.2 Průběh procesu</a> ) a umožňuje vybrat zvolenou Pracovní oblast.	Závazný	1
<b>POB_6.1</b>	Pracovní oblast – zobrazení obsahu	Systém umožňuje uživateli zobrazit obsah vybrané Pracovní oblasti (viz POB_6.0). Zobrazuje seznam všech výkazů s vazbou na vybranou Pracovní oblast.	Závazný	1
<b>POB_7.0</b>	Pracovní oblast – kontrola konzistence	Systém umožňuje uživateli spustit kontrolu konzistence (viz kapitola <a href="#">2.5 Kontrola konzistence</a> ) nad obsahem vybrané Pracovní oblastí.	Závazný	1
<b>POB_7.1</b>	Pracovní oblast – výsledek kontroly konzistence	Systém zobrazuje uživateli seznam Výkazů ve formě tabulky (gridu), s vazbami na Pracovní oblast, u kterých kontrola konzistence skončila chybou. Umožňuje uživateli u těchto 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 nutné vytvořit novou verzi/variantu výkazu.	Závazný	1

### 7.3 Výkaz

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>VYK_1.0</b>	Výkaz - nový	Systém umožňuje uživateli založit první verzi instance objektu Výkaz (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ), který je popsán standardními atributy (viz kapitola <a href="#">3.2.1.1 Atributy objektu Výkaz</a> ) a neobsahuje vazby na žádné verze/varianty Bloku výkazu.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		První verze instance objektu Výkaz vzniká právě v jedné aktivní Pracovní oblasti a má systémem nastaven stav na Projektovaný.		
VYK_1.1	Výkaz - replikace	<p>Systém umožňuje uživateli založit první verzi instance objektu Výkaz jako replikaci již existujícího Výkazu (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a>). Takto založený Výkaz obsahuje Bloky výkazu vytvořené podle BLV_1.1.</p> <p>Nová verze/varianta Výkazu vzniká právě v jedné aktivní Pracovní oblasti a má systémem nastaven stav na Projektovaný.</p>	Závazný	1
VYK_1.2	Výkaz – zařazení do Pracovní oblasti	<p>Systém umožňuje uživateli zařadit do aktivní Pracovní oblasti jakoukoliv verzi/variantu Výkazu jejím označením v seznamu Výkazů (viz VYK_7.0) a jejím přesunutím do vybrané aktivní Pracovní oblasti. Systém:</p> <ul style="list-style-type: none"> <li>• vytváří novou verzi/variantu zařazovaného Výkazu dle POB_3.0; uživatel rozhodne, zda má být vytvořena nová verze nebo varianta,</li> <li>• nevytváří novou verzi/variantu zařazovaného Výkazu, pokud se jedná o přesun mezi aktivními Pracovními oblastmi (viz POB_3.1)</li> <li>• nevytváří novou verzi/variantu výkazu, který je ve stavu Schválený (viz VYK_4.4).</li> </ul> <p>Verze/varianta Výkazu je do aktivní Pracovní oblasti zařazována včetně jemu podřízených instancí objektů Blok výkazu, Datová oblast a Kontrola.</p>	Závazný	1
VYK_1.3	Výkaz – sledování historie	Systém u každé instance objektu Výkaz sleduje jeho historii podle kapitoly <a href="#">3.2 Objekt Výkaz</a> .	Závazný	1
VYK_2.0	Výkaz – atributy nastavené	Systém nastavuje Výkazu vytvořenému podle VYK_1.0 nebo VYK_1.1 následující atributy:	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	systémem	<ul style="list-style-type: none"> <li>• interní identifikátor objektu,</li> <li>• autor objektu (přihlášený uživatel),</li> <li>• datum vytvoření (aktuální datum),</li> <li>• kdo aktualizoval (přihlášený uživatel),</li> <li>• datum a čas aktualizace (aktuální datum),</li> <li>• platnost_od (platnost_od Pracovní oblasti),</li> <li>• platnost_do (platnost_do Pracovní oblasti),</li> <li>• garant (viz OBE_8.0),</li> <li>• konzistence (ano),</li> <li>• vlastník dat (viz VYK_2.6).</li> </ul>		
VYK_2.1	Výkaz – atributy zadávané uživatelem	<p>Systém umožňuje uživateli vyplnit atributy:</p> <ul style="list-style-type: none"> <li>• kód objektu,</li> <li>• název objektu,</li> <li>• popis objektu,</li> <li>• poznámka.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	Závazný	1
VYK_2.2	Výkaz – atributy (nastavené systémem) měněné uživatelem	<p>Systém umožňuje uživateli měnit následující atributy vyplněné podle VYK_2.0:</p> <ul style="list-style-type: none"> <li>• platnost_od (výběrem z kalendáře v rámci vymezení platnosti Pracovní oblasti, v níž je zařazen, viz POB_2.5),</li> <li>• platnost_do (výběrem z kalendáře v rámci vymezení platnosti Pracovní oblasti, v níž je zařazen, viz POB_2.5),</li> <li>• garant (výběrem ze seznamu zaměstnanců ČNB),</li> <li>• konzistence (boolean),</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>vlastník dat (viz VYK_2.6).</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>		
VYK_2.4	Výkaz – atributy měněné systémem	<p>Systém nastavuje v případě změny některého atributu uživatelem podle VYK_2.1 a VYK_2.2 nebo vytvoření nové verze/varianty Výkazu nové hodnoty atributům:</p> <ul style="list-style-type: none"> <li>kdo aktualizoval (uživatel, který objekt vytvořil),</li> <li>datum a čas aktualizace (aktuální datum).</li> </ul>	Závazný	1
VYK_2.5	Výkaz – jednoznačnost atributu kód objektu	<p>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 použita.</p>	Závazný	1
VYK_2.6	Výkaz – atribut vlastník dat	<p>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.</p>	Závazný	2
VYK_2.7	Výkaz – unikátnost atributu název objektu	<p>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.</p>	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
VYK_3.0	Výkaz – vazba na Blok výkazu	System umožňuje uživateli do Výkazu zařadit neomezený počet Bloků výkazu (viz kapitola <a href="#">3.3 Objekt Blok výkazu</a> ).	Závazný	1
VYK_3.1	Výkaz – vazba na Vykazovací rámec	System umožňuje uživateli zařadit Výkaz do neomezeného počtu Vykazovacích rámců. Zařazení Výkazu do Vykazovacího rámce je vymezeno platnost_od a platnost_do. Časová platnost zařazení Výkazu do více Vykazovacích rámců se může vzájemně překrývat.	Závazný	1
VYK_3.2	Výkaz – vazba na Kontrolu	System umožňuje uživateli na Výkaz navázat neomezeně instancí objektu Kontrola (viz kapitola <a href="#">3.20 Objekt Kontrola</a> ).	Závazný	1
VYK_4.0	Výkaz – změna atributů	System umožňuje uživateli měnit atributy verze/varianty Výkazu v souladu se stanovenými pravidly (viz kapitola <a href="#">2.3 Vazby mezi jednotlivými objekty</a> ).	Závazný	1
VYK_4.1	Výkaz – změna vazeb na podřízené objekty	System 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	System 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 oblastí	System 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	System umožňuje uživateli změnit stav instance objektu Výkaz ze stavu Schválený na stav Projektovaný (viz OBE_6.0).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	Projektovaný			
<b>VYK_4.5</b>	Výkaz – zplnění	Systém umožňuje uživateli zplatnit instanci objektu Výkaz (viz OBE_5.0).	Závazný	1
<b>VYK_4.6</b>	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
<b>VYK_5.0</b>	Výkaz – ukončení platnosti uživatelem	Systém umožňuje uživateli ukončit platnost instanci objektu Výkaz (viz OBE_2.0).	Závazný	1
<b>VYK_5.1</b>	Výkaz – ukončení platnosti systémem v závislosti na zplnění jiné verze/varianty téhož objektu	Systém automaticky ukončuje platnost instanci objektu Výkaz (viz OBE_2.1).	Závazný	1
<b>VYK_5.2</b>	Výkaz – prodloužení platnosti uživatelem	Systém umožňuje uživateli prodloužit platnost instance objektu Výkaz (viz OBE_3.0).	Závazný	1
<b>VYK_6.0</b>	Výkaz – smazání	Systém umožňuje uživateli smazat instanci objektu Výkaz (viz OBE_1.0).	Závazný	1
<b>VYK_7.0</b>	Výkaz – zobrazení seznamu	Systém umožňuje uživateli zobrazit seznam obsahující verze/varianty Výkazů ve formě tabulky (grid).	Závazný	1
<b>VYK_7.1</b>	Výkaz – vytvoření	Systém vytváří strukturu verze/varianty Výkazu na základě struktury jemu podřízených instancí objektů Blok výkazu a Datová oblast.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	struktury systémem	Struktura těchto objektů je určena objekty určujícími dimenze Datových oblastí (viz kapitola <a href="#">3.4 Objekt Datová oblast</a> ).		
<b>VYK_7.2</b>	Výkaz – vytvoření struktury uživatelem	Systém umožňuje uživateli vytvořit strukturu verze/varianty Výkazu prostřednictvím funkcionalit rastrového prostředí pro tvorbu Výkazu (viz kapitola <a href="#">5.3.2.2 Rastrové prostředí</a> ).	Závazný	2
<b>VYK_7.3</b>	Výkaz – vytvoření struktury importem	Systém umožňuje uživateli importovat strukturu verze/varianty Výkazu do rastrového prostředí pro tvorbu Výkazu (viz kapitola <a href="#">5.3.2.2 Rastrové prostředí</a> ), kde ji může dále upravovat (viz VYK_7.2).	Závazný	3
<b>VYK_7.4</b>	Výkaz – zobrazení struktury	Systém umožňuje uživateli zobrazit strukturu verze/varianty Výkazu (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ) výběrem ze seznamu (viz VYK_7.0).	Závazný	1
<b>VYK_7.5</b>	Výkaz – export struktury	Systém umožňuje uživateli zobrazenou strukturu verze/varianty Výkazu (viz VYK_7.4) exportovat do formátů DOC, DOCX, XLS, XLSX.	Závazný	2
<b>VYK_7.6</b>	Výkaz – prezentační vrstva	Systém umožňuje uživateli zobrazit prezentační vrstvu verze/varianty Výkazu (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ) výběrem ze seznamu (viz VYK_7.0).	Závazný	2
<b>VYK_7.7</b>	Výkaz – úprava prezentační vrstvy	Systém umožňuje uživateli upravit prezentační vrstvu verze/varianty Výkazu v souladu s kapitolou <a href="#">5.3 Proces tvorby Výkazu</a> )	Závazný	2
<b>VYK_7.8</b>	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
<b>VYK_8.0</b>	Výkaz – prezentace	Systém prezentuje instance objektu Výkaz interním uživatelům v plném rozsahu a historii (viz kapitola <a href="#">5.7 Proces prezentace Výkazu</a> ).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	internímu uživateli	Interním uživatelům jsou prezentovány Výkazy ve všech stavech (Projektovaný, Schválený i Platný).		
<b>VYK_8.1</b>	Výkaz prezentace Osobám	– Systém prezentuje instance objektu Výkaz Osobám v rozsahu a historii určené interním uživatelem (viz kapitola <a href="#">5.7 Proces prezentace Výkazu</a> ). 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 <a href="#">D – Sběr dat, kapitola 5 Formáty pro výměnu dat</a> ) Systém v produkčním prostředí Osobám prezentuje instance objektu Výkaz, které jsou ve stavech Schválený a Platný.	Závazný	2
<b>VYK_8.2</b>	Výkaz předběžná prezentace Osobám	– Systém umožňuje prezentovat Osobám instance objektu Výkaz, které jsou ve stavu Projektovaný prostřednictvím testovacího prostředí.	Závazný	2
<b>VYK_8.3</b>	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 href="#">A – Obecné požadavky, kapitola 2.1.2 Testovací prostředí</a> ).	Závazný	2
<b>VYK_8.4</b>	Výkaz prezentace provedených změn	– Systém umožňuje uživateli vybrat k prezentaci pouze změny provedené v poslední verzi/variantě Výkazu oproti přechodí verzi/variantě Výkazu.	Závazný	2
<b>VYK_9.0</b>	Výkaz – kontrola konzistence	Systém umožňuje uživateli spustit kontrolu konzistence (viz kapitola <a href="#">2.5 Kontrola konzistence</a> ) nad vybranou instancí objektu Výkaz.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
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	<p>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.</p> <p>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.</p>	Závazný	2

#### 7.4 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 <a href="#">5.3 Proces tvorby Výkazu</a> ), který je popsán standardními atributy (viz kapitola <a href="#">3.3.1.1 Atributy objektu Blok výkazu</a> )	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>a neobsahuje vazby na žádné verze varianty Datové oblasti.</p> <p>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.</p>		
<b>BLV_1.1</b>	Blok výkazu – replikace	<p>Systém umožňuje uživateli založit Blok výkazu jako replikaci již existujícího Bloku výkazu (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a>). Takto založený Blok výkazu obsahuje Datové oblasti vytvořené podle DOB_1.1.</p> <p>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.</p>	Závazný	1
<b>BLV_1.2</b>	Blok výkazu – vytvoření verze/varianty uživatelem	Systém umožňuje uživateli vytvořit verzi/variantu jakékoliv instanci objektu Blok výkazu, jenž je v instanci objektu Výkaz, který je ve stavu Projektovaný.	Závazný	1
<b>BLV_1.3</b>	Blok výkazu – vytvoření verze/varianty systémem	<p>Systém vytváří novou verzi/variantu Bloku výkazu, pouze pokud je v instanci objektu Výkaz, který je ve stavu Projektovaný.</p> <p>Systém vytváří novou verzi/variantu instance objektu Blok výkazu (viz kapitola <a href="#">2.3 Vazby mezi jednotlivými objekty</a>), pokud byla uživatelem nebo systémem vytvořena nová verze/varianta jemu podřízeného objektu Datová oblast.</p>	Závazný	1
<b>BLV_1.4</b>	Blok výkazu – sledování historie	Systém u každé instance objektu Blok výkazu sleduje jeho historii podle kapitoly <a href="#">3.3 Objekt Blok výkazu</a> )	Závazný	1
<b>BLV_2.0</b>	Blok výkazu – atributy nastavené	Systém nastavuje Bloku výkazu vytvořenému podle BLV_1.0 nebo BLV_1.1 následující atributy:	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	systémem	<ul style="list-style-type: none"> <li>• interní identifikátor objektu,</li> <li>• autor objektu (přihlášený uživatel),</li> <li>• datum vytvoření (aktuální datum),</li> <li>• kdo aktualizoval (přihlášený uživatel),</li> <li>• datum a čas aktualizace (aktuální datum),</li> <li>• platnost_od (platnost_od výkazu),</li> <li>• platnost_do (platnost_od výkazu),</li> <li>• pořadí (dle pořadí jeho vytvoření v instanci objektu Výkaz).</li> </ul>		
BLV_2.1	Blok výkazu – atributy zadávané uživatelem	<p>Systém umožňuje uživateli vyplnit atributy:</p> <ul style="list-style-type: none"> <li>• kód objektu,</li> <li>• název objektu,</li> <li>• popis objektu,</li> <li>• poznámka.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	Závazný	1
BLV_2.2	Blok výkazu – atributy měněné systémem	<p>Systém nastavuje v případě změny některého atributu uživatelem podle BLV_2.1 nebo vytvoření nové verze/varianty Bloku výkazu nové hodnoty atributům:</p> <ul style="list-style-type: none"> <li>• kdo aktualizoval (uživatel, který objekt vytvořil),</li> <li>• datum a čas aktualizace (aktuální datum).</li> </ul>	Závazný	1
BLV_2.3	Blok výkazu – unikátnost atribut název objektu	<p>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</p>	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		použita pro jiný Blok výkazu v rámci jedné verze/varianty instance objektu Výkaz.		
<b>BLV_2.4</b>	Blok výkazu – atributy (nastavené systémem) měněné uživatelem	Systém umožňuje uživateli změnit atribut pořadí nastavený podle BLV_2.0 za dodržení podmínek stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .	Závazný	1
<b>BLV_2.5</b>	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
<b>BLV_3.0</b>	Blok výkazu – vazba na Datovou oblast	Systém umožňuje uživateli do Bloku výkazu zařadit neomezený počet Datových oblastí (viz kapitola <a href="#">3.3 Objekt Blok výkazu</a> ).	Závazný	1
<b>BLV_4.0</b>	Blok výkazu – změna atributů	Systém umožňuje uživateli měnit atributy verze/varianty Bloku výkazu v souladu se stanovenými pravidly v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .	Závazný	1
<b>BLV_4.1</b>	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
<b>BLV_4.2</b>	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

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>BLV_4.4</b>	Blok výkazu – zplatnění	Systém zplatňuje instanci objektu Blok výkazu v rámci zplatňování jemu nadřazené instance objektu Výkaz (viz VYK_4.2).	Závazný	1
<b>BLV_4.5</b>	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
<b>BLV_5.0</b>	Blok výkazu – ukončení platnosti uživatelem	Systém umožňuje uživateli ukončit platnost instanci objektu Blok výkazu (viz OBE_2.0).	Závazný	1
<b>BLV_5.1</b>	Blok výkazu – ukončení platnosti systémem v závislosti na zplnění jiné verze/varianty téhož objektu	Systém automaticky ukončí platnost instanci objektu Blok výkazu (viz OBE_2.1).	Závazný	1
<b>BLV_5.2</b>	Blok výkazu – prodloužení platnosti	Systém prodlužuje platnost instance objektu Blok výkazu v rámci prodloužení platnosti nadřazené instance objektu Výkaz (viz VYK_5.2).	Závazný	1
<b>BLV_6.0</b>	Blok výkazu – smazání	Systém umožňuje uživateli smazat instanci objektu Blok výkazu (viz OBE_1.0).	Závazný	1
<b>BLV_7.0</b>	Blok výkazu – zobrazení seznamu všech	Systém umožňuje uživateli zobrazit seznam obsahující všechny verze/varianty Bloků výkazů ve formě tabulky (grid).	Závazný	1
<b>BLV_7.1</b>	Blok výkazu – zobrazení	Systém umožňuje uživateli zobrazit seznam obsahující všechny verze/varianty Bloků výkazů jedné instance objektu Výkaz ve formě	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	seznamu podle Výkazu	tabulky (grid).		
<b>BLV_7.2</b>	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 <a href="#">3.4 Objekt Datová oblast</a> ).	Závazný	1
<b>BLV_7.3</b>	Blok výkazu – vytvoření struktury uživatelem	Systém umožňuje uživateli vytvořit strukturu verze/varianty Bloku výkazu prostřednictvím funkcionalit rastrového prostředí pro tvorbu Výkazu (viz kapitola <a href="#">5.3.2.2 Rastrové prostředí</a> ).	Závazný	2
<b>BLV_7.4</b>	Blok výkazu – vytvoření struktury importem	Systém umožňuje uživateli importovat strukturu verze/varianty Bloku výkazu do rastrového prostředí pro tvorbu Výkazu (viz kapitola <a href="#">5.3.2.2 Rastrové prostředí</a> ), kde ji může dále upravovat (viz BLV_7.3).	Závazný	3
<b>BLV_7.5</b>	Blok výkazu – zobrazení struktury	Systém umožňuje uživateli zobrazit strukturu verze/varianty Bloku výkazu (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ) výběrem ze seznamu (viz BLV_7.0 a BLV_7.1).	Závazný	1
<b>BLV_7.6</b>	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
<b>BLV_7.7</b>	Blok výkazu – zobrazení prezentační vrstvy	Systém umožňuje uživateli zobrazit prezentační vrstvu verze/varianty Bloku výkazu (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ) výběrem ze seznamu (viz BLV_7.0 nebo BLV_7.1).	Závazný	2
<b>BLV_7.8</b>	Blok výkazu – úprava prezentační vrstvy	Systém umožňuje uživateli upravit prezentační vrstvu verze/varianty Bloku výkazu v souladu s kapitolou <a href="#">5.3 Proces tvorby Výkazu</a> .	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>BLV_7.9</b>	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
<b>BLV_8.0</b>	Blok výkazu – prezentace internímu uživateli	Systém prezentuje instance objektu Blok výkazu interním uživatelům v plném rozsahu a historii (viz kapitola <a href="#">5.7 Proces prezentace Výkazu</a> ). 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
<b>BLV_8.1</b>	Blok výkazu – prezentace Osobám	Systém prezentuje instance objektu Blok výkazu Osobám pouze v rámci prezentace instance objektu Výkaz (viz VYK_8.1).	Závazný	2

## 7.5 Datová oblast

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>DOB_1.0</b>	Datová oblast - nová	Systém umožňuje uživateli založit první verzi instance objektu Datová oblast (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ), který je popsán standardními atributy (viz kapitola <a href="#">3.4.1 Atributy objektu Datová oblast</a> ) a neobsahuje vazby na žádné verze/varianty objektů popisujících údaje.  První verze instance objektu Datová oblast vzniká právě v jedné instanci objektu Blok výkazu, jenž je ve stavu Projektovaný. Tato verze Datové oblasti má systémem nastaven stav na Projektovaný.	Závazný	1
<b>DOB_1.1</b>	Datová oblast –	Systém umožňuje uživateli založit Datovou oblast jako replikaci již	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	replikace	<p>existující Datové oblasti (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a>). Takto založená instance objektu Datová oblast obsahuje vazby na verze/varianty objektů popisujících Údaje.</p> <p>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.</p>		
<b>DOB_1.2</b>	Datová oblast – klonování	<p>Systém umožňuje uživateli založit Datovou oblast klonováním již existující instance objektu Datová oblast (viz kapitola <a href="#">3.4 Objekt Datová oblast</a>).</p> <p>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.</p> <p>Takto založená instance objektu Datová oblast obsahuje vazby na stejné verze/varianty objektů popisujících Údaje jako mateřská Datová oblast.</p> <p>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“.</p>	Závazný	2
<b>DOB_1.3</b>	Datová oblast – vytvoření verze/varianty uživatelem	Systém umožňuje uživateli vytvořit verzi/variantu jakékoliv instanci objektu Datová oblast, jenž je v instanci objektu Blok výkazu, který je ve stavu Projektovaný.	Závazný	1
<b>DOB_1.4</b>	Datová oblast –	Systém vytváří novou verzi/variantu instance objektu Datová oblast (viz	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	vytvoření verze/varianty systémem	kapitola <a href="#">2.3 Vazby mezi jednotlivými objekty</a> ), pokud byla uživatelem nebo systémem vytvořena nová verze/varianta instancí objektů popisujících Údaje, které jsou v Datové oblasti použity (viz kapitola <a href="#">5.4 Proces tvorby objektů popisujících údaje</a> ).  Systém vytváří novou verzi/variantu Datové oblasti, pouze pokud je v instanci objektu Výkaz, který je ve stavu Projektovaný.		
<b>DOB_1.5</b>	Datová oblast – sledování historie	Systém u každé instance objektu Datová oblast sleduje její historii podle kapitoly <a href="#">3.4 Objekt Datová oblast</a> ).	Závazný	1
<b>DOB_2.0</b>	Datová oblast – atributy nastavené systémem	Systém nastavuje Datové oblasti vytvořené podle DOB_1.0, DOB_1.1 nebo DOB_1.2 následující atributy: <ul style="list-style-type: none"> <li>• interní identifikátor objektu,</li> <li>• autor objektu (přihlášený uživatel),</li> <li>• datum vytvoření (aktuální datum),</li> <li>• kdo aktualizoval (přihlášený uživatel),</li> <li>• datum a čas aktualizace (aktuální datum),</li> <li>• platnost_od (platnost_od bloku výkazu),</li> <li>• platnost_do (platnost_do bloku výkazu),</li> <li>• pořadí (dle pořadí jejího vytvoření v rámci instance objektu Blok výkazu),</li> <li>• garant (viz OBE_8.0),</li> <li>• umožnit sdílení před schválením výkazu (ne).</li> </ul>	Závazný	1
<b>DOB_2.1</b>	Datová oblast – atributy zadávané uživatelem	Systém umožňuje uživateli vyplnit atributy: <ul style="list-style-type: none"> <li>• kód objektu,</li> <li>• název objektu,</li> <li>• popis objektu,</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>• poznámka,</li> <li>• typ datové oblasti.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>		
<b>DOB_2.2</b>	Datová oblast – atributy měněné systémem	<p>Systém nastavuje v případě změny některého atributu uživatelem podle DOB_2.1 nebo vytvoření nové verze/varianty Datové oblasti nové hodnoty atributům:</p> <ul style="list-style-type: none"> <li>• kdo aktualizoval (uživatel, který objekt vytvořil),</li> <li>• datum a čas aktualizace (aktuální datum).</li> </ul>	Závazný	1
<b>DOB_2.3</b>	Datová oblast – atributy (zadané systémem) měněné uživatelem	<p>Systém umožňuje uživateli měnit atributy zadané podle DOB_2.0:</p> <ul style="list-style-type: none"> <li>• garant objektu (výběrem ze seznamu zaměstnanců ČNB),</li> <li>• umožnit sdílení před schválením výkazu,</li> <li>• pořadí,</li> </ul> <p>které jsou nastaveny systémem, za splnění podmínek uvedených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	Závazný	1
<b>DOB_2.4</b>	Datová oblast – unikátnost atributu název objektu	<p>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 oblast v rámci jedné verze/varianty instance objektu Výkaz.</p>	Závazný	1
<b>DOB_2.5</b>	Datová oblast – jednoznačnost atributu kód	<p>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</p>	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	objektu	objektu lze změnit jen v případě, že existuje pouze první verze, a to ve stavu Projektovaný.		
<b>DOB_3.0</b>	Datová oblast – vazba na Údaj	Systém umožňuje uživateli v Datové oblasti vytvořit neomezený počet Údajů (viz kapitola <a href="#">3.4 Objekt Datová oblast</a> ).	Závazný	1
<b>DOB_4.0</b>	Datová oblast – změna atributů	Systém umožňuje uživateli měnit atributy verze/varianty Datové oblasti v souladu s pravidly stanovenými v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .	Závazný	1
<b>DOB_4.1</b>	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
<b>DOB_4.2</b>	Datová oblast – použití objektů popisujících Údaje	Systém umožňuje uživateli v instanci objektu Datová oblast použít neomezený počet objektů popisujících Údaje (viz kapitola <a href="#">3.19 Objekt Údaj</a> ) podle pravidel definovaných v (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ).  Systém umožňuje uživateli vybírat instance objektů popisujících Údaje ze seznamu ve formě tabulky (grid), který je přístupný z okruhu Knihovna (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ).	Závazný	1
<b>DOB_4.3</b>	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 kapitoly <a href="#">5.3.2.1 Formulářové prostředí</a> a <a href="#">5.3.2.2 Rastrové prostředí</a> ).	Závazný	1
<b>DOB_4.4</b>	Datová oblast – informace o nové verzi/variantě	Systém informuje uživatele o vytvoření nové verze/varianty mateřské Datové oblasti (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ).	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	mateřské Datové oblasti			
<b>DOB_4.5</b>	Datová oblast – informace o zařazení Výkazu obsahujícího klonovanou Datovou oblast	System informuje uživatele o zařazení instance objektu Výkaz, jenž obsahuje klonovanou Datovou oblast, do aktivní pracovní oblasti (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ).	Závazný	2
<b>DOB_4.6</b>	Datová oblast – propagace změn v mateřské Datové oblasti do klonů	System automaticky propaguje změny provedené ve verzi/variantě mateřské Datové oblasti do všech jejích klonů, které jsou součástí instancí objektu Výkaz, jež jsou zařazeny do aktivní Pracovní oblasti.	Závazný	2
<b>DOB_4.7</b>	Datová oblast – zrušení vazby mateřské Datové oblasti a klonované Datové oblasti	System zruší všechny vazby modifikované mateřské Datové oblasti na všechny její klony, jež jsou součástí instancí objektů Výkaz, které nejsou zařazeny do žádné aktivní Pracovní oblasti. System zruší všechny vazby klonovaných Datových oblastí na mateřskou Datovou oblast v případě, že uživatel modifikuje klonovanou Datovou oblast (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ).	Závazný	2
<b>DOB_4.8</b>	Datová oblast – schválení	System schvaluje instanci objektu Datová oblast v rámci schvalování jemu nadřizené instance objektu Blok výkazu (viz BLV_4.2) Datovou oblast bez alespoň jednoho Údaje nelze schválit.	Závazný	1
<b>DOB_4.9</b>	Datová oblast – zrušení vazby klonované Datové oblasti a mateřské	System umožňuje uživateli zrušit vazbu vybrané klonované Datové oblasti na její mateřskou Datovou oblast.	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	Datové oblasti uživatelem			
<b>DOB_4.10</b>	Datová oblast – zplatnění	System zplatňuje instanci objektu Datová oblast v rámci zplatňování jemu nadřazené instance objektu Blok výkazu (viz BLV_4.4).	Závazný	1
<b>DOB_4.11</b>	Datová oblast – potlačení vykazování	System 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
<b>DOB_4.12</b>	Datová oblast – vyhledávání duplicit Údajů	System informuje uživatele o vzniku více Údajů se stejnou konkretizací (viz kapitola <a href="#">3.19 Objekt Údaj</a> ) v rámci Datových oblastí, které jsou součástí jedné instance objektu Výkaz (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ).	Závazný	1
<b>DOB_4.13</b>	Datová oblast – změna varianty na verzi	System umožňuje uživateli změnit vytvořenou variantu instance objektu Datová oblast ve stavu Projektovaný na verzi instance objektu Datová oblast (viz OBE_7.0).	Závazný	1
<b>DOB_4.14</b>	Datová oblast – zobrazení údaje na více místech výkazu	System umožňuje uživateli určit u duplicitních údajů (viz DOB_4.12) právě jeden z nich, který bude vykazovaný. System označí automaticky zbylé duplicitní údaje jako zobrazované.	Závazný	1
<b>DOB_5.0</b>	Datová oblast – ukončení platnosti uživatelem	System umožňuje uživateli ukončit platnost instanci objektu Datová oblast (viz OBE_2.0).	Závazný	1
<b>DOB_5.1</b>	Datová oblast – ukončení platnosti systémem v závislosti na zplanění jiné	System automaticky ukončuje platnost instanci objektu Datová oblast (viz OBE_2.1).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	verze/varianty téhož objektu			
<b>DOB_5.2</b>	Datová oblast – prodloužení platnosti	Systém prodlužuje platnost instance objektu Datová oblast v rámci prodloužení platnosti nadřizené instance objektu Blok výkazu (viz BLV_5.2).	Závazný	1
<b>DOB_6.0</b>	Datová oblast – smazání	Systém umožňuje uživateli smazat instanci objektu Datová oblast (viz OBE_1.0).	Závazný	1
<b>DOB_7.0</b>	Datová oblast – zobrazení seznamu všech Datových oblastí	Systém umožňuje uživateli zobrazit seznam obsahující všechny verze/varianty Datových oblastí ve formě tabulky (grid).	Závazný	1
<b>DOB_7.1</b>	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
<b>DOB_7.2</b>	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
<b>DOB_7.3</b>	Datová oblast – vytvoření struktury systémem	Systém vytváří strukturu verze/varianty Datové oblasti na základě použití objektů popisujících Údaje Datové oblasti (viz kapitola <a href="#">5.3.2.1 Formulářové prostředí</a> ). Struktura této Datové oblasti je určena objekty určujícími dimenze Datových oblastí (viz kapitola <a href="#">3.4 Objekt Datová oblast</a> ).	Závazný	1
<b>DOB_7.4</b>	Datová oblast – vytvoření	Systém umožňuje uživateli vytvořit strukturu verze/varianty Datové oblasti prostřednictvím funkcionalit rastrového prostředí pro tvorbu	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	struktury uživatelem	Výkazu, kde strukturu Datové oblasti konkretizuje objekty popisujícími Údaje (viz kapitola <a href="#">5.3.2.2 Rastrové prostředí</a> ).		
<b>DOB_7.5</b>	Datová oblast – vytvoření struktury importem	Systém umožňuje uživateli importovat strukturu verze/varianty Datové oblasti do rastrového prostředí pro tvorbu Výkazu (viz kapitola <a href="#">5.3.2.2 Rastrové prostředí</a> a kapitola <a href="#">6.1 Proces Přebírání metapopisu z externích zdrojů</a> ), kde ji může dále upravovat (viz DOB_7.4). Strukturu Datové oblasti dále uživatel popisuje pomocí objektů popisujících údaje.	Závazný	3
<b>DOB_7.6</b>	Datová oblast – zobrazení struktury	Systém umožňuje uživateli zobrazit strukturu verze/varianty Datové oblasti (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ) výběrem ze seznamu (viz DOB_7.0, DOB_7.1 a DOB_7.2).	Závazný	1
<b>DOB_7.7</b>	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
<b>DOB_7.8</b>	Datová oblast – zobrazení prezentační vrstvy	Systém umožňuje uživateli zobrazit prezentační vrstvu verze/varianty Datové oblasti (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ) výběrem ze seznamu (viz DOB_7.0, DOB_7.1 a DOB_7.2).	Závazný	2
<b>DOB_7.9</b>	Datová oblast – úprava prezentační vrstvy	Systém umožňuje uživateli upravit prezentační vrstvu verze/varianty Datové oblasti v souladu s (viz kapitola <a href="#">5.3 Proces tvorby Výkazu</a> ).	Závazný	2
<b>DOB_7.10</b>	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
<b>DOB_8.0</b>	Datová oblast – prezentace internímu uživateli	Systém prezentuje instance objektu Datová oblast interním uživatelům v plném rozsahu a historii (viz kapitola <a href="#">5.7 Proces prezentace Výkazu</a> ). Tato prezentace je možná v rámci kontextu celé instance objektu Výkaz nebo instance objektu Blok výkazu nebo instance objektu Datová oblast samostatně.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Interním uživatelům jsou prezentovány Datové oblasti ve všech stavech (Projektovaný, Schválený i Platný).		
<b>DOB_8.1</b>	Datová oblast – prezentace Osobám	Systém prezentuje instance objektu Datová oblast Osobám pouze v rámci prezentace instance objektu Výkaz (viz VYK_8.1).	Závazný	1
<b>DOB_11.0</b>	Identifikační Parametr v Datové oblasti	Systém umožňuje uživateli použít na dynamické Datové oblasti Parametr typu identifikační.	Závazný	1
<b>DOB_11.1</b>	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

## 7.6 Číselník, Položka číselníku

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>CIS_1.0</b>	Vytvoření prázdné instance Číselník	Systém umožní vytvoření prázdné instance objektu Číselník podle byznys pravidel definovaných v kapitole <a href="#">3.5 Objekt Číselník</a> . 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	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		v kapitole <a href="#">2.2 Sledování historie objektů.</a>		
CIS_1.1	Jednoznačnost kódu objektu Číselník	System kontroluje, zda hodnota atributu kód objektu nově vytvářené instance objektu Číselník je jednoznačná v porovnání s hodnotami téhož atributu v rámci již existujících instancí objektu Číselník. System nepovoluje změnit kód objektu na hodnotu již použitou u jiné instance objektu Číselník.	Závazný	1
CIS_1.2.	Formát kódu Číselníku	Kód instance objektu Číselník se skládá z prefixu a pořadového čísla: <ul style="list-style-type: none"> <li>• prefix vybírá uživatel nabídkou z listu číselníkových položek, např. BA, SE, viz CIS_1.4,</li> </ul> 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	System 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	System 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	System umožňuje měnit atributy existujícího objektu Číselník za splnění byznys podmínek definovaných v kapitole <a href="#">3.5 Objekt Číselník.</a> 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 <a href="#">2.3 Vazby mezi jednotlivými objekty.</a>	Závazný	1
CIS_3.0	Smazání objektu	System umožňuje smazat objekt Číselník, viz OBE_1.0.	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	Číselník	Smazání objektu instance Číselník je povoleno za následujících podmínek: <ul style="list-style-type: none"> <li>• z Číselníku není vytvořena žádná Hierarchie číselníku,</li> <li>• z Číselníku není vytvořena žádná Doména číselníku,</li> <li>• není použita žádná Položka číselníku pro definování Konkretizovaného ukazatele.</li> </ul>		
<b>CIS_4.0</b>	Ukončení platnosti objektu Číselník	Systém umožňuje ukončit platnost instanci objektu (viz OBE_2.0 a OBE_2.1).	Závazný	1
<b>CIS_5.0</b>	Prodloužení platnosti instance objektu Číselník	Systém umožňuje instanci objektu Číselník prodloužit platnost (viz OBE_3.0).	Závazný	1
<b>CIS_6.0</b>	Schválení instance objektu Číselník	Systém umožňuje schválit instanci objektu Číselník (viz OBE_4.0).	Závazný	1
<b>CIS_7.0</b>	Zplatnění instance objektu Číselník	Systém umožňuje zplatnit instanci objektu Číselník (viz OBE_5.0).	Závazný	1
<b>CIS_8.0</b>	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
<b>CIS_9.0</b>	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
<b>CIS_11.0</b>	Dynamické	Systém umožňuje uživateli definovat další atributy Číselníku podle	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	atributy objektu Číselník	byznys pravidel definovaných v kapitole <a href="#">3.6.2 Dynamické atributy objektu Položka číselníku</a> . Takto přidané atributy jsou zároveň atributem každé Položky číselníku.		
<b>CIS_12.0</b>	Atribut pro správu objektu	Systém umožní uživateli nastavit atribut na požadovanou hodnotu a přiřadit danému Číselníku požadované uživatelské místo. Podrobněji viz kapitola <a href="#">3.5.1 Atributy objektu Číselník</a> .	Závazný	2
<b>CIS_13.0</b>	Vytvoření Položky číselníku	Systém umožní uživateli vytvořit novou Položku číselníku objektu Číselník podle pravidel definovaných v kapitole <a href="#">3.6 Objekt Položka číselníku</a> .	Závazný	1
<b>CIS_13.1</b>	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
<b>CIS_14.0</b>	Vytvoření Položky číselníku ručním typováním	Systém umožní uživateli přidávat položky do objektu Číselník ve stavu Projektovaný ručním typováním.	Závazný	1
<b>CIS_15.0</b>	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. Požadované formáty:	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		a) XLS – atributy jako hlavičky sloupců v pořadí odpovídající datovému modelu objektu Položka číselníku, b) XML – atributy jako XML elementy.		
<b>CIS_16.0</b>	Import Položek číselníku z externího souboru	<p>Systém umožňuje uživateli vložit nebo aktualizovat Položky číselníku prostřednictvím souboru vytvořeného podle CIS_15.0.</p> <p>Atributem, přes který probíhá porovnání obsahu souboru a obsahu objektu Položka číselníku, je kód položky.</p> <p>Import má dvě fáze:</p> <ol style="list-style-type: none"> <li>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:             <ol style="list-style-type: none"> <li>i. položky, které budou přidány,</li> <li>ii. položky, které budou aktualizovány,</li> <li>iii. položky, které budou zrušeny,</li> <li>iv. zda číselník obsahuje hierarchii položek.</li> </ol> </li> <li>2) samotný import: uživatel má možnost se rozhodnout, jaké operace (i. – iv.) importem souboru provede:             <ol style="list-style-type: none"> <li>i. přidání nových Položek číselníku –ano/ne,</li> <li>ii. aktualizaci hodnot atributů (kromě klíčového atributu kód položky) – ano/ne,</li> <li>iii. zrušení položek – ano/ne, ukončit platnost/smazat,</li> <li>iv. vytvoření Hierarchie číselníku – ano/ne.</li> </ol> </li> </ol> <p>V případě vytváření Hierarchie číselníku systém vytvoří instanci Hierarchie číselníku s generickým názvem a následně vytvoří Položky hierarchie na základě vztahů mezi Položkami číselníku v importovaném souboru.</p>	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>Bod iii lze provést za podmínek definovaných v požadavku CIS_18.0.</p> <p>Bod iv. lze provést, pouze pokud byly provedeny všechny předchozí body. Při tomto způsobu vytváření Hierarchie číselníku musí být dodržena všechna pravidla podle kapitoly <a href="#">7.7 Hierarchie číselníku, Položka hierarchie</a>.</p>		
CIS_17.0	Změna Položky číselníku	Systém umožňuje změnit atributy u existující Položky číselníku podle pravidel definovaných v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>	Závazný	1
CIS_18.0	Smazání Položky číselníku	<p>Systém umožní smazat Položku číselníku podle pravidel uvedených v kapitole <a href="#">2.4.5.4 Smazání objektů</a>. Smazání Položky číselníku je povoleno jen za následujících podmínek:</p> <ul style="list-style-type: none"> <li>• Číselník musí být ve stavu Projektovaný,</li> <li>• Položka číselníku nesmí být použita v Konkretizovaném parametru,</li> <li>• Položka číselníku nesmí být použita v Převodníku,</li> <li>• Položka číselníku nesmí být použita v žádné Doméně číselníku,</li> <li>• nesmí existovat vazba Položky číselníku na Položku hierarchie.</li> </ul>	Závazný	1
CIS_19.0	Ukončení platnosti Položky číselníku	<p>Systém umožní uživateli ukončit Položku číselníku podle pravidel uvedených v kapitole <a href="#">2.4.5.3 Ukončování platnosti objektů</a>.</p> <p>Ukončení platnosti Položky číselníku je povoleno za stejných podmínek jako v CIS_12.0.</p>	Závazný	1
CIS_20.0	Označení součtové Položky číselníku v Číselníku	Systém označí součtovou Položku číselníku v seznamu Položek číselníku znakem $\Sigma$ v závislosti na tom, zda existuje vazba na uzlovou položku objektu Hierarchie číselníku.	Závazný	1
CIS_21.0	Zobrazení elementárního	Systém umožní uživateli zobrazit elementární Položky číselníku pro součtovou Položku číselníku přímo ze seznamu Položek číselníku.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	rozkladu součtové Položky číselníku v Číselníku			
<b>CIS_22.0</b>	Základní zobrazení Položek číselníku	System umožní uživateli zobrazit pořadí Položek číselníku v základním abecedně řazeném seznamu podle atributu kód položky.	Závazný	1
<b>CIS_23.0</b>	Uživatelské řazení Položek číselníku	System 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 země, 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
<b>CIS_24.0</b>	Export Položek číselníku	System umožňuje uživateli vyexportovat Položky číselníku, zobrazené dle CIS_22.0 nebo CIS_23.0, do zvoleného formátu (xls, pdf, xml).	Závazný	2
<b>CIS_25.0</b>	Tisk Číselníku	System umožní uživateli tisk zobrazeného Číselníku, a to v pořadí podle požadavku C_22.0 nebo C_23.0.	Závazný	2
<b>CIS_26.0</b>	Volitelné parametry tisku Číselníku s položkami	System umožní uživateli nastavit parametricky, které atributy mají být vytištěny.	Závazný	2
<b>CIS_27.0</b>	Hlídaní duplicit	System neumožní vytvoření duplicitní Položky číselníku v rámci jednoho Číselníku (duplicitní kód i název).	Závazný	1

## 7.7 Hierarchie číselníku, Položka hierarchie

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>HIE_1.0</b>	Vytvoření Hierarchie číselníku	<p>Systém umožňuje vytvořit novou prázdnou instanci objektu Hierarchie číselníku za podmínek definovaných v kapitole <a href="#">3.7 Objekt Hierarchie číselníku</a>.</p> <p>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 <a href="#">2.2.6 Přístup „Sledování historie – časová platnost + stavy“</a>.</p> <p>Systém kontroluje název a kód vytvářené Hierarchie číselníku. Pokud indikuje shodu s již existující Hierarchií číselníku, nepovolí uživateli vytvořit novou instanci Hierarchie číselníku.</p>	Závazný	1
<b>HIE_1.0</b>	Jednoznačnost kódu objektu Hierarchie číselníku	<p>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.</p>	Závazný	1
<b>HIE_2.0</b>	Editace atributů Hierarchie číselníku	<p>Systém umožňuje měnit hodnoty atributů existující instance Hierarchie číselníku za splnění byznys podmínek uvedených v kapitole <a href="#">3.7 Objekt Hierarchie číselníku</a>.</p> <p>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 <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p> <p>Pokud není změna atributu v nějakém stavu možná, pak systém uživateli tuto změnu vůbec nedovolí.</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Tento funkční požadavek se nevztahuje k editaci „atributu pro správu objektu“ a „atributu pro provedení synchronizace“.		
<b>HIE_3.0</b>	Nastavení atributu pro správu Hierarchie číselníku	Nastavení „atributu pro správu objektu“ je zděděno od zdrojového Číselníku.	Závazný	2
<b>HIE_4.0</b>	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
<b>HIE_5.0</b>	Smazání instance objektu Hierarchie číselníku	Systém umožňuje smazat instanci objektu Hierarchie číselníku (viz OBE_1.0).	Závazný	1
<b>HIE_6.0</b>	Plnění objektu Hierarchie číselníku Položkami hierarchie	<p>Systém umožňuje uživateli plnit nebo změnit obsah instance objektu Hierarchie číselníku za následujících podmínek:</p> <ul style="list-style-type: none"> <li>• 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),</li> <li>• všechny ostatní Hierarchie číselníku, které vycházejí ze stejného Číselníku jako vytvářená/měněná Hierarchie číselníku, mají nastaven „atribut provedení synchronizace“ na hodnotu „ano“</li> </ul> <p>Podrobně je proces tvorby instance objektu Hierarchie číselníku popsán v kapitole <a href="#">3.7.2 Proces tvorby Hierarchií číselníku</a>.</p>	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>HIE_7.0</b>	Aktualizace Hierarchie číselníku	<p>Proces aktualizace instance objektu Hierarchie je popsán podrobně v kapitole <a href="#">3.7.2.1 Aktualizace instance Hierarchie číselníku</a>.</p> <p>Systém povolí uživateli provést aktualizaci, pokud má objekt založenu verzi objektu ve stavu Projektovaný.</p> <p>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.</p>	Závazný	1
<b>HIE_8.0</b>	Synchronizace Hierarchie číselníku	<p>Proces synchronizace je možné spustit až po úspěšném dokončení procesu aktualizace.</p> <p>Proces synchronizace instance objektu Hierarchie číselníku je popsán podrobně v kapitole <a href="#">3.7.2.2 Synchronizace instance Hierarchie číselníku</a>.</p>	Závazný	1
<b>HIE_9.0</b>	Pravidla pro řízení aktualizace a synchronizace Hierarchie číselníku	<p>Proces aktualizace a synchronizace je řízen pravidly definovanými v příloze <a href="#">8.2 Příloha 2 - Příklady pro pravidla aktualizace a synchronizace Hierarchií číselníku</a> vč. vzorových příkladů.</p> <p>Chování systému podle pravidel při aktualizaci a synchronizaci je řízeno algoritmy zabudovanými do systému.</p>	Závazný	1
<b>HIE_10.0</b>	Diagnostika dopadů při aktualizaci Hierarchie číselníku	<p>Systém zobrazuje uživateli diagnostiku dopadu změn na všechny související objekty v důsledku změny prováděné v objektu Hierarchie číselníku.</p> <p>V rámci diagnostiky jsou uživateli zobrazovány následující informace:</p> <ul style="list-style-type: none"> <li>• dopad na přímo související objekty, tj. Domény číselníku, které jsou vytvořeny přímo z aktualizované Hierarchie,</li> <li>• dopad na nepřímo související objekty, tj. objekty které používají</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		přímo související objekty (např. Datová oblast, kde je použita Doména číselníku vytvořená z aktualizované Hierarchie číselníku),		
<b>HIE_10.1</b>	Diagnostika dopadů při synchronizaci Hierarchie číselníku	<p>Systém zobrazuje uživateli diagnostiku dopadu změn na všechny související objekty v důsledku změny prováděné v objektu Hierarchie číselníku.</p> <p>V rámci diagnostiky jsou uživateli zobrazovány následující informace:</p> <ul style="list-style-type: none"> <li>• dopad na přímo související objekty, tj. na ostatní Hierarchie číselníku v souvisejících uzlech s vyznačením dopadu do měněného objektu,</li> <li>• dopad na Domény číselníku, které jsou vytvořeny z Hierarchií číselníku, které budou změnou v aktualizované Hierarchii číselníku dotčeny,</li> <li>• dopad na nepřímo související objekty, které obsahují měněné přímo související objekty s vyznačením dopadu do měněných objektů a dopadu na změnu stavu dotčeného objektu.</li> </ul>	Závazný	1
<b>HIE_11.0</b>	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
<b>HIE_12.0</b>	Diagnostika konfliktů při synchronizaci Hierarchie číselníku systémem neřešitelný konflikt	<p>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 očekávaný dopad na přímo související objekty a vyzve uživatele k vyřešení konfliktu.</p> <p>Situace, kdy je vyhodnocen systémem neřešitelný konflikt, jsou popsány v kapitole <a href="#">8.2.9 Konflikt typu Duplicita v nezávislých uzlech</a> resp. <a href="#">8.2.10 Konflikt typu Duplicita v závislých uzlech</a>.</p> <p>V případě obou výše uvedených situací není synchronizace dokončena a</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		uživateli je zobrazena informace o tom, že situace nelze vyřešit.		
<b>HIE_13.0</b>	Diagnostika konfliktů při synchronizaci Hierarchie číselníku systémem řešitelný konflikt	Pokud systém při procesu synchronizace zjistí, že nastala situace, že položka má změnit svojí nadřazenou položku, systém uživateli umožní potvrdit změnu, pokud je požadována, a synchronizace je dokončena.	Závazný	1
<b>HIE_14.0</b>	Propagace změn do souvisejících objektů při aktualizaci	<p>Systém propaguje změny způsobené aktualizací instance objektu Hierarchie číselníku do přímo souvisejících objektů následovně:</p> <ul style="list-style-type: none"> <li>• do objektů Knihovny, Pracovní oblasti a Úložiště platných a schválených Výkazů za podmínky, že platnost dotčených objektů leží v intervalu platnost_od Hierarchie číselníku až platnost_do Hierarchie číselníku,</li> <li>• pro objekty Knihovny, když platnost_od Hierarchie číselníku je rovna platnost_od verze Domény číselníku: <ul style="list-style-type: none"> <li>• jestliže je v rámci aktualizace dotčena změnou Doména číselníku, která je ve stavu Projektovaný, systém provede její změnu okamžitě,</li> <li>• jestliže je v rámci aktualizace dotčena Doména číselníku ve stavu Platný, systém vytvoří Doméně číselníku novou verzi a provede změnu v nové verzi.</li> <li>• jestliže je v rámci aktualizace dotčena Doména ve stavu Schválený, systém vrátí doménu do stavu Projektovaný a provede změnu.</li> </ul> </li> <li>• pro objekty Knihovny, když platnost_od Hierarchie číselníku je větší než platnost_od verze Domény číselníku je Doméně číselníku vždy</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>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.</p> <p>Pro objekty v Pracovní oblasti platí:</p> <ul style="list-style-type: none"> <li>• do objektů v Pracovní oblasti, které jsou ve stavu Projektovaný, jsou změny vyplývající ze změn v aktualizované Hierarchii číselníku promítány okamžitě a uživatel objektu je o tomto informován,</li> <li>• do objektů v Pracovní oblasti, které jsou ve stavu Schválený, nejsou změny promítány okamžitě, uživateli je zaslána informace a uživatel má povinnost avizované změny provést, jinak nastává stav nekonzistence,</li> <li>• do objektů používaných v Pracovní oblasti, které jsou ve stavu Platný (jedná se o Domény číselníku a Ukazatele) např. při přesunu Výkazu do Pracovní oblasti, jsou změny promítány tak, že je nadřizovým objektům vytvořena nová verze a změna je promítnuta (není-li již vytvořena).</li> </ul> <p>Pro objekty v Úložišti platných a schválených výkazů platí:</p> <ul style="list-style-type: none"> <li>• objekty v Úložišti platných a schválených výkazů, nejsou změnou dotčeny okamžitě. Uživateli je zaslána informace a uživatel se rozhodne, zda změnu promítne,</li> <li>• pokud změna není promítnuta, pak kontrola konzistence ohlásí nesoulad.</li> </ul>		
<b>HIE_15.0</b>	Propagace změn do souvisejících objektů při synchronizaci	<p>Pro objekty Knihovny platí totéž jako v HIE_14.0 pro Domény číselníku.</p> <p>Pro související Hierarchie číselníku platí následující:</p> <ul style="list-style-type: none"> <li>• do souvisejících Hierarchií číselníku ve stavu Projektovaný jsou změny promítány okamžitě,</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>• související Hierarchie číselníku ve stavu schváleném jsou vráceny do stavu Projektovaný a změny jsou promítnuty,</li> <li>• 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.</li> </ul> <p>Pro objekty v pracovní oblasti platí totéž jako v HIE_14.0.</p> <p>Pro objekty v Úložišti platných a schválených výkazů platí totéž jako v HIE_14.0.</p>		
<b>HIE_16.0</b>	Prostředí pro projektování Hierarchie číselníku	<p>Systém umožní grafické prostředí pro projektování objektu Hierarchie číselníku, v němž má uživatel možnost:</p> <ul style="list-style-type: none"> <li>• opticky rozlišit jednotlivé úrovně Hierarchie číselníku,</li> <li>• označovat souvislé části Hierarchie číselníku,</li> <li>• vytvářet Hierarchie číselníku z Položek hierarchie ostatních Hierarchií číselníku,</li> <li>• přesouvat celé uzle z jedné úrovně do druhé,</li> <li>• vkládat Položky číselníku pod označenou Položku hierarchie na stejnou úroveň nebo o úroveň níže,</li> <li>• mazat celé uzle Hierarchie číselníku.</li> </ul>	Závazný	1
<b>HIE_16.1</b>	Podpůrné funkce pro projektování Hierarchie číselníku	<p>Systém umožňuje v rámci uživatelského prostředí používat následující funkce:</p> <ul style="list-style-type: none"> <li>• vyhledávání Položek hierarchie v určité Hierarchii číselníku,</li> <li>• možnost zobrazení jen určité úrovně,</li> <li>• tisk Hierarchie číselníku s grafickým vyznačením úrovní,</li> <li>• zobrazení souvisejících uzlů v ostatních Hierarchiích číselníku,</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>• zobrazení diagnostiky dopadu při aktualizaci a synchronizaci Hierarchie číselníku,</li> <li>• export diagnostiky dopadu do textového souboru.</li> </ul>		
<b>HIE_17.0</b>	Smazání instance Hierarchie číselníku včetně jejího obsahu tj. instancí Položek hierarchie	System umožní 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
<b>HIE_18.0</b>	Ukončení platnosti Hierarchie číselníku	System umožní ukončení platnosti instance objektu Hierarchie (viz OBE_2.0 a OBE_2.1).	Závazný	1
<b>HIE_19.0</b>	Zobrazení Hierarchie číselníku	System umožňuje uživateli zobrazit Hierarchii číselníku tak, aby byly graficky rozlišeny jednotlivé úrovně.	Závazný	1
<b>HIE_20.0</b>	Export Hierarchie číselníku	System umožňuje uživateli exportovat celou Hierarchii číselníku do souboru ( pdf, xls, xml).	Závazný	2
<b>HIE_21.0</b>	Tisk Hierarchie číselníku	System umožňuje vytištění celé Hierarchie číselníku a jejích atributů včetně informace o verzi zdrojového Číselníku, ze kterého je Hierarchie číselníku vytvořena.	Závazný	2
<b>HIE_23.0</b>	Atribut pro provedení synchronizace	System 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.  V seznamu Hierarchií číselníku jsou označeny všechny synchronizované	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>Hierarchie číselníku. Nesynchronizovaná Hierarchie číselníku je opticky rozlišena.</p> <p>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.</p>		
<b>HIE_24.0</b>	Hlídaní duplicit	Systém neumožní uživateli vytvořit instanci objektu Hierarchie číselníku se stejným kódem a názvem v rámci jednoho Číselníku.	Závazný	1
<b>HIE_25.0</b>	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
<b>HIE_26.0</b>	Prodloužení platnosti instance objektu Hierarchie číselníku	Systém umožňuje uživateli prodloužit platnost instance objektu Hierarchie číselníku (viz OBE_3.0).	Závazný	1
<b>HIE_27.0</b>	Schválení instance objektu Hierarchie číselníku	Systém umožňuje uživateli schválit instanci objektu Hierarchie číselníku (viz OBE_4.0).	Závazný	1



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

## 7.8 Doména číselníku

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>DOC_1.0</b>	Vytvoření prázdné instance ruční Doména číselníku	<p>System umožňuje vytvoření instance objektu Doména číselníku podle byznys pravidel definovaných v kapitole <a href="#">3.9 Objekt Doména číselníku</a>.</p> <p>Při vytvoření nové instance je automaticky založena nová verze Domény číselníku. Podrobně je popis životního cyklu popsán v <a href="#">2.2.6 Přístup „Sledování historie – časová platnost + stavy“</a>.</p> <p>Vytvoření prázdné instance je prováděno uživatelem v případě ruční Domény číselníku.</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DOC_2.0	Vytvoření automatické Domény číselníku	<p>Systém umožňuje vytvoření automatické Domény číselníku z Hierarchie číselníku a to následovně:</p> <ul style="list-style-type: none"> <li>• 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,</li> <li>• systém takto vytvořené doméně přiřadí automaticky standardní atributy a vyplní je následovně: <ul style="list-style-type: none"> <li>○ název objektu - je stejný jako název uzlové položky,</li> <li>○ kód objektu – je odvozen od kódu uzlové položky a to následovně A_+“kód uzlové položky“_pořadové číslo,</li> <li>○ datum_vytvoření - datum, kdy uživatel Doménu číselníku vytvořil,</li> <li>○ platnost_od a platnost_do - je odvozena od zdrojové instance objektu Hierarchie číselníku,</li> </ul> </li> <li>• ostatní atributy nejsou systémem plněny. Možnost jejich vyplnění je na uživateli,</li> <li>• ostatní základní atributy jsou nastavovány dle popisu uvedených v kapitole <a href="#">2.4.2 Standardní atributy objektů</a>.</li> </ul>	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
DOC_3.0	Editace hodnot atributů ruční Domény číselníku	Systém umožňuje měnit hodnoty atributů existujících instancí objektu ruční Doména číselníku za splnění byznys podmínek uvedených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		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 <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .		
<b>DOC_4.0</b>	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
<b>DOC_5.0</b>	Nastavení atributu Typ Domény číselníku	Systém eviduje podle způsobu vytvoření instance objektu Doména číselníku její typ, tj. <b>automatická</b> nebo <b>ruční</b> a podle evidovaného typu povolí úpravy hodnot atributů podle funkčních požadavků DOC_4.0 nebo DOC_3.0.  Nastavení hodnoty atributu viz kapitola <a href="#">3.9 Objekt Doména číselníku</a>	Závazný	1
<b>DOC_6.0</b>	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 <a href="#">3.9.1 Atributy objektu Doména číselníku</a>	Závazný	1
<b>DOC_7.0</b>	Hlídaní použití domény při projektování	Systém podle typu Domény číselníku hlídá použití Domény číselníku pro projektování výkazů, a to následovně: a) ruční Doména číselníku: <ul style="list-style-type: none"> <li>• lze použít pro projektování statických a kartotékových Datových oblastí,</li> <li>• pro dynamické Datové oblasti je použití ruční Domény číselníku omezeno jen na konkretizaci instance objektu Ukazatele, tj. Konkretizovaný ukazatel,</li> </ul> b) automatická Doména číselníku - lze použít pro projektování	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		dynamických Datových Oblastí.		
DOC_8.0	Duplicitní Domény číselníku	System 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	System umožňuje uživateli zobrazit: a) seznam elementárních Položek číselníku v plochem rozlišení, b) seznam součtových i elementárních Položek číselníku v plochem 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	System modifikuje při aktualizaci instance objektu Doména číselníku vytvořené z Hierarchie číselníku Domény číselníku podle pravidel popsanych v čísloných položkách 1 až 10 v kapitole <a href="#">3.9 Objekt Doména číselníku</a> a příloze <a href="#">8.3 Příloha 3 - Příklady pro pravidla aktualizace Domén číselníku</a> .	Závazný	1
DOC_11.0	Smazání ruční Domény číselníku	System umožňuje uživateli smazat Doménu číselníku (viz OBE_1.0).	Závazný	1
DOC_12.0	Smazání automatické Domény číselníku	System umožňuje smazání uživatelem vytvořené automatické Domény číselníku. Toto smazání nemá žádný dopad na Hierarchii číselníku, ze které automatická Doména číselníku vznikla.  Smazání je umožněno za podmínky, že automatická Doména číselníku není použita pro konkretizaci Parametru v dynamické Datové oblasti, který tvoří dimenzi.	Závazný	1
DOC_13.0	Ukončení	System umožňuje ukončení platnosti ruční domény podle byznys pravidel	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	platnosti ruční Domény číselníku	uvedených v OBE_2.0 a OBE_2.1.		
<b>DOC_14.0</b>	Ukončení platnosti automatické Domény	Systém umožňuje ukončení platnosti automatické domény číselníku (viz OBE_2.0 a OBE_2.1).	Závazný	1
<b>DOC_17.0</b>	Tisk Domény číselníku	<p>Systém umožňuje uživateli vytisknout jednotlivou Doménu číselníku, skupinu Domén číselníku nebo všechny Domény číselníku podle parametrů nastavených na začátku tisku. Parametry tisku jsou:</p> <ul style="list-style-type: none"> <li>• Číselník, více Číselníků, všechny Číselníky,</li> <li>• interval časové platnosti,</li> <li>• jednotlivé Domény číselníku, více Domén číselníku, všechny Domény číselníku.</li> </ul> <p>Pro parametrizaci tiskových sestav systém umožní použít tzv. „wild cards“.</p> <p>Na tiskové sestavě je uveden zdrojový Číselník, verze Domény číselníku a interval její platnosti.</p>	Závazný	2
<b>DOC_18.0</b>	Export Domény číselníku	Systém umožní export Domény číselníku do textového souboru ve zvoleném parametrickém rozsahu jako DOC_17.	Závazný	2
<b>DOC_19.0</b>	Prodloužení platnosti instance objektu Doména číselníku	Systém umožňuje uživateli prodloužit platnost instance objektu Doména číselníku (viz OBE_3.0).	Závazný	1
<b>DOC_20.0</b>	Schválení instance objektu	Systém umožňuje uživateli převést instanci objektu Doména číselníku ze stavu Projektovaný do stavu Schválený (viz OBE_4.0).	Závazný	1

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

## 7.9 Převodník, Položka převodníku

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
--------------	-----------------	-----------------	------------	-----------

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
PŘE_1.0	Zobrazení Převodníků a Převodníkových položek	<p>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).</p> <p>Systém v seznamu Převodníků automaticky zobrazuje nejaktuálnější verzi a variantu instance objektu Převodník, nicméně umožňuje získat informace i o všech jeho předcházejících verzích/variantách.</p> <p>Ze zobrazení jsou patrné zejména následující informace:</p> <ul style="list-style-type: none"> <li>• zobrazení kompletní historie všech verzí a variant, včetně vymezení časové oblasti každé verze/varianty, kterými daný Převodník prošel během svého životního cyklu,</li> <li>• zobrazení kompletní sady stavů, kterými prošla ta která verze/varianta Převodníku,</li> <li>• pro každý Převodník je zobrazeno, z jakých číselníků je tvořen (vazby zdrojový číselník/cílový číselník v objektovém modelu),</li> <li>• pro každou Položku převodníku je zobrazeno, z jakých číselníkových položek je tvořena (vazba zdrojová položka/cílová položka v objektovém modelu).</li> </ul>	Závazný	3
PŘE_2.0	Vytvoření Převodníku	<p>Systém umožňuje vytvořit nový Převodník (instanci objektu Převodník) za splnění byznys podmínek uvedených v kapitole <a href="#">3.10 Objekt Převodník</a>.</p> <p>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 <a href="#">2.2 Sledování historie instancí objektů</a></p>	Závazný	3
PŘE_2.1	Jednoznačnost	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené	Závazný	3



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	kódu objektu Převodník	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 použita.		
<b>PŘE_3.0</b>	Editace Převodníku	<p>Systém umožňuje měnit atributy existujícího Převodníku (existující instance objektu Převodník) za splnění byznys podmínek uvedených v kapitole <a href="#">3.10 Objekt Převodník</a>.</p> <p>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 <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	Závazný	3
<b>PŘE_4.0</b>	Smazání Převodníku	<p>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.</p> <p>Smazání instance objektu Položka převodníku nemá žádný vliv na instance objektu Položka číselníku.</p>	Závazný	3
<b>PŘE_5.0</b>	Ukončení časové platnosti Převodníku	<p>Systém umožňuje ukončit časovou platnost vybrané verze Převodníku v souladu s funkčním požadavkem OBE_2.0.</p> <p>Podrobně je postup ukončování platnosti verze popsán v <a href="#">2.2.6 Přístup „Sledování historie – časová platnost + stavy“</a>.</p> <p>Zároveň musí dojít k ukončení časové platnosti všech podřízených</p>	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		souvisejících instancí objektu Položka převodníku tak, aby jejich časová platnost nevybočila z rozsahu nadřazené instance objektu Převodník.		
<b>PŘE_6.0</b>	Vytvoření Položky převodníku	<p>Systém umožňuje vytvořit novou Položku převodníku v rámci vybraného převodníku za splnění byznys podmínek definovaných v kapitole <a href="#">3.11 Objekt Položka převodníku</a>.</p> <p>Vytvořit novou Položku převodníku je možno bez ohledu na to, v jakém stavu se nachází nadřazená instance objektu Převodník. Je však nutné, aby systém zohlednil nastavení závislosti mezi objekty (viz kapitola <a href="#">2.3 Vazby mezi jednotlivými objekty</a>), a podle toho buď vytvořil novou verzi nebo variantu nebo neprovedl žádný zásah do existující instance objektu Převodník.</p>	Závazný	3
<b>PŘE_7.0</b>	Změna Položky převodníku	<p>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 <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p> <p>Změnit existující položku převodníku je možno bez ohledu na to, v jakém stavu se nachází nadřazená instance objektu Převodník, je však nutné, aby systém zohlednil nastavení závislosti mezi objekty (viz kapitola <a href="#">2.3 Vazby mezi jednotlivými objekty</a>), a podle toho buď vytvořil novou verzi nebo variantu nebo neprovedl žádný zásah do existující instance objektu Převodník.</p> <p>Součástí změny Položky převodníku může být změna časové platnosti (nastavení hodnot atributů <code>platnost_od</code> a <code>platnost_do</code>). Při změně této časové platnosti musí být zajištěno splnění pravidla, že časová platnost instance objektu Položka převodníku nesmí vybočovat z časové platnosti nadřazené související instance objektu Převodník</p>	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>PŘE_8.0</b>	Smazání Položky převodníku	<p>Systém umožňuje smazat existující Položku převodníku bez ohledu na jakékoli další okolnosti.</p> <p>Smazat existující položku převodníku je možno bez ohledu na to, v jakém stavu se nachází nadřazená instance objektu Převodník, je však nutné, aby systém zohlednil nastavení závislosti mezi objekty (viz kapitola <a href="#">2.3 Vazby mezi jednotlivými objekty</a>), a podle toho buď vytvořil novou verzi nebo variantu nebo neprovedl žádný zásah do existující instance objektu Převodník.</p>	Závazný	3
<b>PŘE_9.0</b>	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
<b>PŘE_10.0</b>	Zplatnění nové verze/varianty Převodníku	Systém umožňuje zplatnění nové verze Převodníku v souladu s pravidly definovanými v požadavku OBE_5.0.	Závazný	3

### 7.10 Účtová osnova, účet

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>UCO_1.0</b>	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
<b>UCO_2.0</b>	Vytvoření účtové	Systém umožňuje uživateli vytvořit novou Účtovou osnovu (první	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	osnovy	instanci objektu Účtová osnova) za splnění byznys podmínek uvedených v kapitole <a href="#">3.12 Objekt Účtová osnova</a> . Při vytvoření nové Účtové osnovy systém automaticky vytvoří novou verzi a automaticky tuto verzi založí ve stavu Projektovaný.		
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 použita.	Závazný	1
UCO_3.0	Editace Účtové osnovy	Systém umožňuje měnit atributy existující Účtové osnovy (existující instance) za splnění byznys podmínek uvedených v kapitole <a href="#">3.12 Objekt Účtová osnova</a> .  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 osnovou.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>Pokud je mazaný Účet použit, dojde ke zrušení vazby tohoto Účtu na Ukazatele (a Konkretizované ukazatele).</p> <p>Pokud jsou použity Účty, které nejsou mazány, vazba zůstává zachována.</p>		
UCO_5.0	Ukončení platnosti Účtové osnovy	<p>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 (a Konkretizované ukazatele) při dodržení podmínek závislosti (viz kapitola <a href="#">2.3.3</a> Objekt #ObjektZávislost).</p>	Závazný	1
UCO_5.1	Ukončení platnosti Účtové osnovy systémem	<p>Systém automaticky ukončuje platnost verzi/variantě objektu Účtová osnova, jehož následující verzi/variantě byl uživatelem změněn stav Schválený na Platný (viz OBE_2.1).</p>	Závazný	1
UCO_6.0	Prodloužení platnosti Účtové osnovy	<p>Systém umožňuje uživateli prodloužit platnost Účtové osnovy včetně Účtů v ní obsažených (viz OBE_3.0).</p>	Závazný	1
UCO_7.0	Změna obsahu Účtové osnovy	<p>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.</p>	Závazný	1
UCO_8.0	Schválení Účtové osnovy	<p>Systém umožňuje uživateli schválit instanci objektu Účtová osnova (viz OBE_4.0).</p>	Závazný	1
UCO_9.0	Zplatnění účtové osnovy	<p>Systém umožňuje uživateli zplatnit instanci objektu Účtová osnova (viz OBE_5.0).</p>	Závazný	1
UCO_10.0	Vrácení Účtové osnovy do	<p>Systém umožňuje převést instanci objektu Účtová osnova ze stavu Schválený do stavu Projektovaný (viz OBE_6.0).</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	projekce			
UCO_21.0	Vytvoření účtu	System umožňuje uživateli vytvořit Účet v rámci definované Účtové osnovy za splnění byznys podmínek definovaných v kapitole <a href="#">3.13 Objekt Účet</a> .	Závazný	1
UCO_21.1	Jednoznačnost kódu objektu Účet	System 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 použita.	Závazný	1
UCO_22.0	Editace Účtu	System umožňuje měnit atributy existujícího Účtu za splnění byznys podmínek uvedených v kapitole <a href="#">3.13 Objekt Účet</a> . System 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	System 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 (a Konkretizované ukazatele).	Závazný	1
UCO_24.0	Ukončení platnosti Účtu	System umožňuje ukončit časovou platnost Účtu za splnění byznys podmínek definovaných v kapitole <a href="#">3.13 Objekt Účet</a> . S ukončením platnosti Účtu je současně v odpovídajícím časovém řezu zrušena vazba daného Účtu na Ukazatele (a Konkretizované ukazatele) při dodržení podmínek závislosti (viz kapitola <a href="#">2.3.3 Objekt #ObjektZávislost</a> ).	Závazný	1
UCO_25.0	Prodloužení	System umožňuje prodloužit platnost Účtu, který byl ukončen, za	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	platnosti Účtu	podmínky, že poslední instance Účtové osnovy má platnost_do vyšší nebo rovnu platnosti_do Účtu.		
UCO_31.0	Použití Účtové osnovy	System umožňuje uživateli zobrazit Ukazatele (a Konkretizované ukazatele), které mají definovanou vazbu na Účty z dané Účetní osnovy (viz kapitola <a href="#">2.4.1 Základní vlastnosti objektů</a> ).	Závazný	1
UCO_32.0	Historie Účtové osnovy	System umožňuje uživateli zobrazit historii Účtové osnovy (viz KNH_3.0).	Závazný	1
UCO_33.0	Použití Účtu	System umožňuje uživateli zobrazit Ukazatele (a Konkretizované ukazatele), které mají vazbu na vybraný Účet (viz <a href="#">2.4.1 Základní vlastnosti objektů</a> ).	Závazný	1
UCO_34.0	Historie Účtu	System umožňuje uživateli zobrazit historii Účtu (viz KNH_3.0).	Závazný	1
UCO_41.0	Zobrazení vazby na Účet ze vzoru Výkazu	System umožňuje uživateli zobrazit v grafické struktuře Výkazu souhrnný přehled k jednotlivým Konkretizovaným ukazatelům vazbu na Účet.	Závazný	2

### 7.11 Datový typ

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DAT_1.0	Zobrazení Datového typu	System 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 <a href="#">3.14 Objekt Datový typ</a> .	Závazný	1
DAT_2.0	Vytvoření	System umožňuje uživateli vytvořit nový Datový typ (první instanci	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	Datového typu	objektu Datový typ) za splnění byznys podmínek uvedených v kapitole <a href="#">3.14 Objekt Datový typ</a> . Při vytvoření nového Datového typu systém automaticky vytvoří novou verzi a automaticky tuto verzi založí ve stavu Projektovaný.		
<b>DAT_2.1</b>	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
<b>DAT_2.2.</b>	Formát kódu Datový typ	Kód instance objektu Datový typ se skládá z prefixu a dalších znaků: <ul style="list-style-type: none"> <li>• prefix vybírá uživatel nabídkou z listu číselníkových položek, např. n, p, R, viz DAT_2.4,</li> </ul> k vybranému prefixu uživatel přiřazuje další alfanumerické znaky nebo podtržítko.	Závazný	1
<b>DAT_2.3</b>	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
<b>DAT_2.4</b>	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
<b>DAT_3.0</b>	Editace Datového typu	Systém umožňuje měnit atributy existujícího Datového typu (existující instance) za splnění byznys podmínek uvedených v kapitole <a href="#">3.14 Objekt Datový typ</a> . 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	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		prováděné změny jsou v souladu s nastavením systému, které definuje, jak mohou být jednotlivé atributy měněny.		
<b>DAT_3.1</b>	Změna varianty Datového typu ve verzi	Systém umožňuje uživateli změnit variantu Datového typu ve verzi (viz OBE_7.0).	Závazný	1
<b>DAT_4.0</b>	Smazání Datového typu	Systém umožňuje uživateli smazat Datový typ ve stavu Projektovaný, pokud není nikde použit (viz OBE_1.0).	Závazný	1
<b>DAT_4.1</b>	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
<b>DAT_5.0</b>	Ukončení platnosti Datového typu	Systém umožňuje uživateli ukončit platnost Datového typu, pokud není nikde použit (viz OBE_2.0).	Závazný	1
<b>DAT_5.1</b>	Ukončení platnosti Datového typu včetně DDT	Systém umožňuje uživateli ukončit platnost Datového typu včetně souvisejících Domén datového typu, pokud nejsou nikde použity.	Závazný	1
<b>DAT_5.2</b>	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
<b>DAT_6.0</b>	Prodloužení platnosti Datového typu	Systém umožňuje uživateli prodloužit platnost Datového typu (viz OBE_3.0).	Závazný	1
<b>DAT_7.0</b>	Schválení	Systém umožňuje uživateli schválit instanci objektu Datový typ (viz	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	Datového typu	OBE_4.0).		
<b>DAT_8.0</b>	Zplatnění Datového typu	System umožňuje uživateli zplatnit instanci objektu Datový typ (viz OBE_5.0).	Závazný	1
<b>DAT_9.0</b>	Vrácení Datového typu do projekce	System umožňuje převést instanci objektu Datový typ ze stavu Schválený do stavu Projektovaný (viz OBE_6.0).	Závazný	1
<b>DAT_10.0</b>	Použití Datového typu	System umožňuje uživateli zobrazit použití Datového typu (viz kapitola <a href="#">2.4.1 Základní vlastnosti objektů</a> ).	Závazný	1
<b>DAT_11.0</b>	Historie Datového typu	System umožňuje uživateli zobrazit historii Datového typu (viz KNH_3.0).	Závazný	1

### 7.12 Doména datového typu

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>DDT_1.0</b>	Zobrazení Domény datového typu	System 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 <a href="#">3.15 Objekt Doména datového typu</a> .	Závazný	1
<b>DDT_2.0</b>	Vytvoření Domény datového typu	System umožňuje uživateli vytvořit novou Doménu datového typu (první instanci objektu Doména Datového typu), za splnění byznys podmínek uvedených v kapitole <a href="#">3.15 Objekt Doména datového typu</a> . Při vytvoření nové Domény datového typu systém automaticky vytvoří novou verzi a automaticky tuto verzi založí ve stavu Projektovaný.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
DDT_2.1	Jednoznačnost kódu objektu Doména datového typu	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené 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.	Závazný	1
DDT_3.0	Editace Domény datového typu	Systém umožňuje měnit atributy existující Domény datového typu (existující instance) za splnění byznys podmínek uvedených v kapitole <a href="#">3.15 Objekt Doména datového typu</a> . 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

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>DDT_6.0</b>	Prodloužení platnosti Domény datového typu	System umožňuje uživateli prodloužit platnost Domény datového typu (viz OBE_3.0).	Závazný	1
<b>DDT_7.0</b>	Schválení Domény datového typu	System umožňuje uživateli schválit instanci objektu Domény datového typu (viz OBE_4.0).	Závazný	1
<b>DDT_8.0</b>	Zplatnění Domény datového typu	System umožňuje uživateli zplatnit instanci objektu Domény datového typu (viz OBE_5.0).	Závazný	1
<b>DDT_9.0</b>	Vrácení Domény datového typu do projekce	System umožňuje převést instanci objektu Domény datového typu ze stavu Schválený do stavu Projektovaný (viz OBE_6.0).	Závazný	1
<b>DDT_10.0</b>	Použití Domény datového typu	System umožňuje uživateli zobrazit použití Domény datového typu (viz kapitola <a href="#">2.4.1 Základní vlastnosti objektů</a> ).	Závazný	1
<b>DDT_11.0</b>	Historie Domény datového typu	System umožňuje uživateli zobrazit historii Domény datového typu (viz KNH_3.0).	Závazný	1

### 7.13 Ukazatel, Konkretizace ukazatele, Dodatečná konkretizace ukazatele

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>UKZ_1.0</b>	Zobrazení Ukazatele a jeho detailů	System umožňuje zobrazení náhledu na instance objektu Ukazatel. Součástí náhledu je jednak zobrazení všech atributů instance objektu Ukazatel a další podrobnosti, které vyplývají z napojení objektu Ukazatel na ostatní objekty systému.	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>System v seznamu Ukazatelů automaticky zobrazuje nejaktuálnější verzi a variantu instance objektu Ukazatel, nicméně umožňuje získat informace i o všech jeho předcházejících verzích/variantách.</p> <p>Pro vybraný Ukazatel tak systém v rámci náhledu prezentuje zejména tyto informace:</p> <ul style="list-style-type: none"> <li>• zobrazení kompletní historie všech verzí a variant, včetně vymezení časové oblasti každé verze/varianty, kterými daný Ukazatel prošel během svého životního cyklu,</li> <li>• zobrazení kompletní sady stavů, kterými prošla ta která verze/varianta Ukazatele,</li> <li>• 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 <a href="#">3.13 Objekt Účet</a>,</li> <li>• zobrazení informací o všech Datových oblastech, kde je daný Ukazatel použit (asociační třída Konkretizovaný ukazatel). Pro každou kombinaci Ukazatel/Datová oblast zobrazeno zejména: <ul style="list-style-type: none"> <li>• 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 nadřazeném Ukazateli (je-li definován),</li> <li>• 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</li> </ul> </li> </ul>		

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>objektu Datový Typ, Doména (Doména číselníku nebo Doména datového typu), Položka (Položka číselníku nebo Položka hierarchie) nebo Hierarchie číselníku. Instance objektu Konkretizace ukazatele může být napojena na právě jednu instanci právě jednoho z výše uvedených objektů; XOR,</p> <ul style="list-style-type: none"> <li>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 <a href="#">3.19.3 Dodatečná konkretizace ukazatele (nespolečné Parametry)</a>. Informace o tom, s jakými Účty je Ukazatel do dané Datové oblasti zařazen (objekt Konkretizovaný Ukazatel, atribut Detailní rozpis účtů).</li> </ul>		
UKZ_2.0	Vytvoření Ukazatele	<p>Systém umožňuje vytvořit nový Ukazatel (instanci objektu Ukazatel) za splnění byznys podmínek uvedených v kapitole <a href="#">3.16 Objekt Ukazatel</a>.</p> <p>V rámci definice nového Ukazatele je možné definovat vazbu Ukazatele na Účet. Jedná se o vytvoření instance(i) objektu Zařazení účtu k ukazateli. Účty je možno k Ukazateli přiřazovat za splnění pravidel, která jsou popsána v kapitole <a href="#">3.13 Objekt Účet</a>.</p> <p>V rámci definice ukazatele <b>neprobíhá</b> vytváření instance objektu Konkretizovaný ukazatel. Tato instance vzniká v okamžiku zařazování Ukazatele do Datové oblasti a je popsána ve funkčním požadavku UKZ_6.0.</p> <p>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 <a href="#">2.2 Sledování historie instancí objektů metapopisu</a>.</p>	Závazný	1
UKZ_2.1	Jednoznačnost	Systém kontroluje, zda hodnota atributu kód objektu nově vytvářené	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	kódu objektu Ukazatel	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 objektu Ukazatel.		
<b>UKZ_2.2.</b>	Formát kódu Ukazatele	Kód instance objektu Ukazatel se skládá z prefixu a pořadového čísla: <ul style="list-style-type: none"> <li>• prefix vybírá uživatel nabídkou z listu číselníkových položek, např. EPR, SOLV, viz UKZ_2.4,</li> <li>• 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).</li> </ul>	Závazný	1
<b>UKZ_2.3</b>	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
<b>UKZ_2.4</b>	Prefix kódu objektu Ukazatel	Systém umožňuje uživateli aktualizovat (zakládat nové, ukončovat platnost a mazat nepoužité položky) číselník prefixů kódů pro objekt Ukazatel.	Závazný	1
<b>UKZ_3.0</b>	Editace Ukazatele	Systém umožňuje měnit atributy existujícího Ukazatele (existující instance objektu Ukazatel) za splnění byznys podmínek uvedených v kapitole <a href="#">3.16 Objekt Ukazatel</a> .  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 <a href="#">2.3 Vazby</a>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<a href="#">mezi jednotlivými objekty.</a>		
UKZ_4.0	Smazání Ukazatele	<p>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 Konkretizovaný ukazatel) a v souladu s funkčním požadavkem OBE_1.0.</p> <p>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.</p> <p>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.</p>	Závazný	1
UKZ_5.0	Ukončení platnosti Ukazatele	<p>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).</p> <p>Podrobně je postup ukončování platnosti verze popsán v kapitole <a href="#">2.2.6 Přístup „Sledování historie – časová platnost + stavy“</a>.</p>	Závazný	1
UKZ_6.0	Vytvoření Konkretizovanéh o ukazatele	<p>Systém umožňuje vytvoření nového Konkretizovaného ukazatele (instance objektu Konkretizovaný ukazatel), a to tak, že vybrané Datové oblasti přiřadí vybraný Ukazatel. V rámci přiřazení Ukazatele do Datové oblasti je nutné definovat tzv. <b>pořadí Ukazatele v Datové oblasti</b>. Tato informace je důležitá z hlediska následného procesu generování Údaje (viz kapitola <a href="#">7.15 Údaj</a>, UDJ_1.0 a další tamtéž).</p> <p>Systém umožňuje zařazovat Ukazatele do Datové oblasti hierarchicky. Zachycení vazby nadřizenosti/podřizenosti jednotlivých ukazatelů zajišťuje vazba nadřizený Konkretizovaný ukazatel. Pro zařazení</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>Ukazatele do Datové oblasti platí tato pravidla:</p> <ul style="list-style-type: none"> <li>• 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 Konkretizovaných ukazatelů,</li> <li>• uživatel může při zařazování Ukazatele do Datové oblasti určit, že daný Ukazatel se má stát nadřazeným nebo podřazeným již zařazenému Ukazateli,</li> <li>• pokud má Konkretizovaný ukazatel vazbu na nadřazený Konkretizovaný ukazatel, má právě jednu takovou vazbu (jeden Konkretizovaný ukazatel nemůže mít dva a více přímých nadřazených Konkretizovaných ukazatelů),</li> <li>• pokud uživatel vytváří hierarchii Konkretizovaných ukazatelů, pak musí u každého Konkretizovaného ukazatele, který má nějaké podřazené Konkretizované ukazatele určit, zda je výčet podřazených Konkretizovaných položek úplný. Systém tuto informaci uchová takto: <ul style="list-style-type: none"> <li>- nastaví hodnotu atributu „suma“ na „ano“ v případě, že uživatel určí, že daný Konkretizovaný ukazatel obsahuje úplný výčet podřazených Konkretizovaných ukazatelů,</li> <li>- nastaví hodnotu atributu „suma“ na „ne“, v případě, že uživatel určí, že výčet podřazených Konkretizovaných ukazatelů není úplný.</li> </ul> </li> </ul> <p>Atribut „suma“, resp. jeho hodnota má následně vliv na tzv. automaticky generované kontroly (viz JVK_2.0).</p> <p>Akci je možno provést pouze v případě, že existují instance objektů</p>		

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>Datová oblast a Ukazatel.</p> <p>Vytvořit nový Konkretizovaný ukazatel je možno pouze v případě, že Datová oblast, v rámci které má být Konkretizovaný ukazatel vytvořen, je ve stavu Projektovaný.</p> <p>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).</p> <p>V rámci procesu zařazování Ukazatele do Datové oblasti musí uživatel definovat další informace (viz kapitola <a href="#">3.16 Objekt Ukazatel</a>).</p>		
<b>UKZ_6.1.</b>	Vytvoření Dodatečné konkretizace ukazatele	<p>Systém umožňuje ke každému Konkretizovanému ukazateli definovat tzv. dodatečnou konkretizaci. Jedná se o provázání existující instance objektu Konkretizovaný ukazatel s jednou nebo více instancemi objektu Parametr přes asociační třídu Dodatečná konkretizace ukazatele.</p> <p>Vytvořit novou instanci objektu Dodatečná konkretizace ukazatele je možno pouze v případě, že Datová oblast, v rámci které je Konkretizovaný ukazatel dodatečně konkretizován, je ve stavu Projektovaný.</p> <p>Systém zajišťuje vytvoření Dodatečné konkretizace ukazatele v souladu se všemi byznys podmínkami popsány v kapitole <a href="#">3.19.3 Dodatečná konkretizace ukazatele (nespolečné Parametry)</a></p>	Závazný	1
<b>UKZ_7.0</b>	Změna Konkretizovanéh o ukazatele	<p>Systém umožňuje změnit existující Konkretizovaný ukazatel pouze v případě, že Datová oblast, ke které je existující Ukazatel přes objekt Konkretizovaný ukazatel připojen, je ve stavu Projektovaný.</p> <p>Změnou Konkretizovaného ukazatele se myslí:</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>• <b>změna pořadí Ukazatele v Datové oblasti:</b> uživatel může změnit pořadí jednotlivých Konkretizovaných ukazatelů,</li> <li>• <b>změna nadřazeného Konkretizovaného ukazatele:</b> uživatel může změnit nadřazený Konkretizovaný ukazatel, jedná se o tyto akce: <ul style="list-style-type: none"> <li>○ uživatel smaže odkaz na nadřazený Konkretizovaný ukazatel, tzn., že se z takového Konkretizovaného ukazatele stane Konkretizovaný ukazatel nejvyšší úrovně,</li> <li>○ uživatel přidá informaci o nadřazeném Konkretizovaném ukazateli tomu Konkretizovanému ukazateli, který doposud žádný nadřazený Konkretizovaný ukazatel neměl. Z Konkretizovaného ukazatele nejvyšší úrovně se stane podřazený Konkretizovaný ukazatel,</li> <li>○ uživatel změní informaci o nadřazeném Konkretizovaném ukazateli, tzn., že se daný Konkretizovaný ukazatel přesune pod jiný nadřazený Konkretizovaný ukazatel. Pokud má daný Konkretizovaný ukazatel nějaké podřazené Konkretizované ukazatele jsou i tyto přesunuty pod jiný nadřazený Konkretizovaný ukazatel,</li> </ul> </li> <li>• <b>změna typu konkretizace:</b> uživatel určí, že daný Ukazatel 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 ukazatel nezaniká a tím pádem nezaniká ani k ní napojená instance objektu Dodatečná konkretizace ukazatele, pokud existuje),</li> <li>• <b>změna ukazatele:</b> 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</li> </ul>		

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<p>struktury Konkretizovaných ukazatelů, systém zajišťuje, že nedojde k rozpadu této hierarchie. Výsledkem celé operace je tak výměna jednoho Konkretizovaného ukazatele za jiný.</p>		
<b>UKZ_8.0</b>	Smazání Konkretizovaného ukazatele	<p>Systém umožňuje smazat existující Konkretizovaný ukazatel pouze v případě, že Datová oblast, ke které je existující Ukazatel přes objekt Konkretizovaný ukazatel připojen, je ve stavu Projektovaný.</p> <p>V případě, že dojde ke smazání instance objektu Konkretizovaný ukazatel, jsou smazány i všechny související instance objektu Dodatečná konkretizace ukazatele, pokud nějaké existují.</p> <p>Pokud je mazaný Konkretizovaný ukazatel nadřizený jinému Konkretizovanému ukazateli (vazba „nadřizený Konkretizovaný Ukazatel“), pak systém upozorní na tuto skutečnost uživatele a nabídne tyto možnosti:</p> <ul style="list-style-type: none"> <li>• smazání daného Konkretizovaného ukazatele, včetně smazání všech jeho podřizených Konkretizovaných ukazatelů v celé hloubce stromu,</li> <li>• přesunutí přímých podřizených Konkretizovaných ukazatelů na nejvyšší úroveň,</li> <li>• přesunutí přímých podřizených položek pod jiný Konkretizovaný ukazatel.</li> </ul> <p>Smazání existující instance objektu Konkretizovaný ukazatel nemá žádný vliv na související instance objektu Datová oblast a Ukazatel; ty zůstanou zachovány beze změny.</p>	Závazný	1
<b>UKZ_9.0</b>	Změna Dodatečné konkretizace ukazatele	<p>Systém umožňuje změnit existující dodatečnou konkretizaci Ukazatele (instanci objektu Dodatečná konkretizace ukazatele) pouze v případě, že související Konkretizovaný ukazatel je připojen k Datové oblasti, která je ve stavu Projektovaný.</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		V rámci změny lze provést pouze změnu typu dodatečné konkretizace, tedy určení toho, zda je instance objektu Dodatečná konkretizace ukazatele napojena na Položku číselníku/hierarchie, Doménu číselníku/datového typu, Datový typ nebo Hierarchii číselníku.		
<b>UKZ_10.0</b>	Smazání Dodatečné konkretizace Ukazatele	<p>Systém umožňuje smazat existující instanci objektu Dodatečná konkretizace ukazatele pouze v případě, že související Konkretizovaný ukazatel je připojen k Datové oblasti, která je ve stavu Projektovaný.</p> <p>Smazání instance objektu Dodatečná konkretizace ukazatele nemá vliv na nadřizenou instanci objektu Konkretizovaný ukazatel, ani na objekt Parametr; ty zůstanou zachovány beze změny.</p>	Závazný	1
<b>UKZ_11.0</b>	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
<b>UKZ_12.0</b>	Zplatnění nové verze/varianty Ukazatele	Systém umožňuje zplatnění nové verze Ukazatele v souladu s pravidly definovanými v OBE_5.0	Závazný	1
<b>UKZ_13.0</b>	Ukončení platnosti instance objektu Ukazatel	Systém umožňuje ukončení platnosti instance objektu Ukazatel v souladu s FP OBE_2.1.	Závazný	1
<b>UKZ_14.0</b>	Schválení instance objektu Ukazatel	Systém umožňuje schválit instanci objektu Ukazatel v souladu s FP OBE_4.0	Závazný	1
<b>UKZ_15.0</b>	Hromadné vytvoření instancí	Systém umožňuje hromadně vytvořit instance objektu Ukazatel (za splnění všech omezujících podmínek definovaných pro individuální	Závazný	3



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	objektu Ukazatel	vytvoření instance objektu Ukazatel) v souladu s procesem Tvorba objektů popisujících údaje, viz kapitola <a href="#">5.4 Proces tvorby objektů popisujících údaje</a> ), tj. označení oblasti, která obsahuje Ukazatele a následného předvyplnění (název Ukazatele) formuláře pro hromadné vytvoření Ukazatele, kde bude moci uživatel doplnit další atributy Ukazatele.		

#### 7.14 Parametr, Konkretizovaný parametr

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>PAR_1.0</b>	Zobrazení Parametru a jeho detailů	<p>Systém umožňuje zobrazení náhledu na instance objektu Parametr. Součástí náhledu je jednak zobrazení všech atributů instance objektu Parametr a další podrobnosti, které vyplývají z napojení objektu Parametr na ostatní objekty systému.</p> <p>Systém v seznamu Parametrů automaticky zobrazuje nejaktuálnější verzi/variantu instance objektu Parametr, nicméně umožňuje získat informace i o všech jeho předcházejících verzích/variantách.</p> <p>Pro vybraný Parametr tak systém v rámci náhledu prezentuje zejména tyto informace:</p> <ul style="list-style-type: none"> <li>• 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,</li> <li>• zobrazení kompletní sady stavů, kterými prošla ta která verze/varianta Parametru,</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>• 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 <a href="#">3.18 Objekt Parametr</a>.</li> <li>• zobrazení informací o všech Datových oblastech, kde je daný Parametr použit (asociační třída Konkretizovaný parametr). Pro každou kombinaci Parametr/Datová oblast zobrazeno zejména: <ul style="list-style-type: none"> <li>• 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 Datové oblasti připojen, a informaci o pořadí daného Parametru v rámci dané osy.</li> <li>• 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,</li> </ul> </li> <li>• 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 <a href="#">3.19.3 Dodatečná konkretizace ukazatele (nespolečné Parametry)</a>.</li> <li>• zobrazení seznamu všech Vykazovaných parametrů (objekt Vykazovaný parametr), které se váží k instanci objektu Parametr.</li> </ul>		

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
PAR_2.0	Vytvoření Parametru	<p>Systém umožňuje vytvořit nový Parametr (instanci objektu Parametr) za splnění byznys podmínek uvedených v kapitole <a href="#">3.18 Objekt Parametr</a>.</p> <p>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).</p> <p>V rámci definice Ukazatele <b>neprobíhá</b> vytváření instance objektu Konkretizovaný parametr ani instance objektu Dodatečná konkretizace ukazatele. Tyto instance vznikají v okamžiku zařazování Parametru do 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).</p> <p>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 <a href="#">2.2 Sledování historie instancí objektů</a>.</p>	Závazný	1
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	<p>Kód instance objektu Parametr se skládá z prefixu a pořadového čísla:</p> <ul style="list-style-type: none"> <li>• prefix vybírá uživatel nabídkou z listu číselníkových položek, např. P, I, viz PAR_2.4,</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		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).		
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 instance objektu Parametr) za splnění byznys podmínek uvedených v kapitole <a href="#">3.18 Objekt Parametr</a> .  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 <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .	Závazný	1
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

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>PAR_5.0</b>	Ukončení platnosti Parametru	<p>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).</p> <p>Postup ukončování platnosti verze/varianty popsán v kapitole <a href="#">2.2.6 Přístup „Sledování historie – časová platnost + stavy“</a>.</p>	Závazný	1
<b>PAR_6.0</b>	Vytvoření Konkretizovanéh o parametru	<p>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 <a href="#">3.18 Objekt Parametr</a>.</p> <p>V rámci vzniku nového Konkretizovaného parametru je třeba:</p> <ul style="list-style-type: none"> <li>• 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),</li> <li>• 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),</li> <li>• určit pořadí, v jakém bude Parametr na danou osu zařazen (určení pořadí je nutné jen pro Parametry, které tvoří dimenzi),</li> <li>• určit, na jakou kartu Datové oblasti bude Parametr zařazen (pouze v případě, že se jedná o kartotékovou datovou oblast).</li> </ul> <p>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.</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
PAR_7.0	Změna Konkretizovanéh o parametru	<p>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.</p> <p>Změnou Konkretizovaného parametr se myslí:</p> <ul style="list-style-type: none"> <li>• <b>změna typu konkretizace:</b> 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),</li> <li>• <b>změna osy:</b> uživatel určí, že má být Parametr zařazen na jinou osu Datové oblasti, než je aktuálně zařazen,</li> <li>• <b>změna pořadí:</b> uživatel určí, že Parametr má být do Datové oblasti zařazen v jiném pořadí než je aktuálně zařazen,</li> <li>• <b>změna Parametru:</b> 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).</li> </ul>	Závazný	1
PAR_8.0	Smazání Konkretizovanéh o parametru	<p>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.</p> <p>Smazání existující instance objektu Konkretizovaný parametr nemá žádný vliv na související instance objektu Datová oblast a Parametr; ty zůstanou zachovány beze změny.</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
PAR_9.0	Změna stavu Parametru ze Schválený na Projektovaný	System 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	System 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	System umožňuje ukončení platnosti instance objektu Parametr v souladu s funkčním požadavkem OBE_2.1.	Závazný	1
PAR_12.0	Schválení instance objektu Parametr	System umožňuje schválit instanci objektu Parametr v souladu s funkčním požadavkem OBE_4.0	Závazný	1

### 7.15 Údaj

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
UDJ_1.0	Vytvoření Údaje – generování systémem	<p>System umožňuje vytvořit v rámci projektované Datové oblasti nový Údaj podle následujícího postupu:</p> <ul style="list-style-type: none"> <li>• uživatel přiřadí do Datové oblasti minimálně jeden Ukazatel. Přiřazení Ukazatele do Datové oblasti je popsáno v UKZ_2.0 (objekt Konkretizovaný ukazatel),</li> <li>• uživatel přiřadí do Datové oblasti minimálně jeden Parametr. Přiřazení Parametru do Datové oblasti je popsáno v PAR_2.0,</li> </ul>	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>• uživatel použije akci „hromadné generování údajů“. Systém: <ul style="list-style-type: none"> <li>○ požádá uživatele o informaci, jakou má nastavit údajům citlivost. Uživatel vybere právě jednu citlivost, ze seznamu možných hodnot (viz kapitola <a href="#">3.19.5 Atributy objektu Údaj</a>), tato citlivost pak bude nastavena všem Údajům, které budou v dalším kroku vygenerovány. Uživatel ji pak bude moci individuálně upravit (viz UDJ_2.0),</li> <li>○ zkontroluje, zda v Datové oblasti již existují nějaké Údaje. Pokud ano, pak je všechny smaže,</li> <li>○ vygeneruje sadu nových instancí objektů Údaj podle algoritmu popsaného v kapitole <a href="#">3.19.1 Vznik Údaje – statická Datová oblast v rámci procesu generování</a>. Systém kontroluje, zda v daném Výkazu v jiné Datové oblasti již neexistuje Údaj, který by byl stejně popsán (je mu přiřazen stejný Konkretizovaný ukazatel a stejná sada Konkretizovaných parametrů. V případě, že to tak bude, pak systém: <ol style="list-style-type: none"> <li>1) zjistí identifikátor Údaje, který již existuje v jiné Datové oblasti a je definován stejně jako nově vznikající Údaje,</li> <li>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.</li> </ol> </li> </ul> </li> </ul>		
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	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		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.		
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	<p>Systém umožňuje k vybrané Datové oblasti zobrazit seznam všech Údajů, které jsou v rámci ní vygenerovány. Toto zobrazení je možné bez ohledu na to, v jakém stavu se Datová oblast nachází.</p> <p>Zobrazení Údajů v Datové oblasti probíhá následovně:</p> <ol style="list-style-type: none"> <li>1) systém nejdříve vykreslí strukturu Datové oblasti. Struktura Datové oblasti je vykreslena na základě souvisejících instancí objektů Konkretizovaný Ukazatel a Konkretizovaný Parametr tak, že se zohlední atributy „pořadí“ a „osa“ (v případě Parametru),</li> <li>2) výsledkem operace z bodu 1) je struktura (grid) odpovídající souřadnicím Údajů, neboli jednotlivé buňky Datové oblasti, které představují jednotlivé Údaje,</li> <li>3) systém zobrazí jako neaktivní (například šedou barvou), ty buňky, které představují Údaj, který je označen jako „nevykazovaný“,</li> </ol> <p>Systém umožňuje uživateli označit jakoukoli buňku výsledného gridu a zobrazit její tzv. diagnostiku. Pod pojmem diagnostika buňky rozumíme:</p> <ul style="list-style-type: none"> <li>• informaci o Ukazateli, který je danému Údaji přiřazen,</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>informaci o seznamu všech společných Parametrů, které jsou k Údaji přiřazeny (Parametry společné pro všechny Údaje v Datové oblasti),</li> <li>informaci o seznamu všech nespolečných Parametrů (Parametry přiřazené pouze tomuto konkrétnímu Údaji).</li> </ul>		
<b>UDJ_4.0</b>	Výběr jednoho z duplicitních údajů jako vykazovaného	<p>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.</p> <p>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).</p>	Závazný	1
<b>UDJ_5.0</b>	Namapování Údaje na XBRL Datapoint	Systém umožňuje uživateli provést mapování mezi instancí objektu Údaj a odpovídajícím Datapointem v XBRL taxonomii. Pro případ, že se Údaj nachází ve Výkazu, který odpovídá části existující XBRL taxonomie a je žádoucí umožnit vykazovat data do systému SDAT i prostřednictvím formátu XBRL.	Závazný	2

### 7.16 Jednovýkazová kontrola (JVK)

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>JVK_1.0</b>	JVK – sémantický jazyk JVK	Systém umožňuje uživateli zapsat JVK prostřednictvím sémantického jazyka pro tvorbu kontrol (viz kapitola <a href="#">3.20.3.1 Kontroly vytvořené sémantickým jazykem</a> ).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
JVK_1.1	JVK – algoritmická	System umožňuje uživateli vytvořit algoritmickou JVK (viz kapitola <a href="#">3.20.3.2 Algoritmické kontroly</a> ) 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	System u každé instance objektu JVK sleduje její historii podle kapitoly <a href="#">3.20.3 Objekt Jednovýkazová kontrola (JVK)</a> .	Závazný	1
JVK_1.3	JVK – vytvoření algoritmu dodavatelem	System je dodán včetně algoritmů pro JVK uvedených v kapitole <a href="#">8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systémem SDAT</a> .	Závazný	2
JVK_2.0	JVK – generované systémem	<p>System umožňuje uživateli spustit generování JVK na základě vazeb použitých instancí objektů popisujících Údaje v rámci jedné instance objektu Výkaz (viz kapitola <a href="#">3.20.3 Objekt Jednovýkazová kontrola (JVK)</a>) v souladu s postupem definovaným v kapitole <a href="#">5.5.2.1 Věcné kontroly generované automaticky systémem</a>.</p> <p>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).</p> <p>System defaultně generuje JVK pro všechny Údaje instance objektu Výkaz, pro které lze tyto kontroly vygenerovat.</p> <p>System generuje kontroly pouze v sémantickém jazyce (viz JVK_1.0) a pouze pro instanci objektu Výkaz, jenž se nachází ve stavu Projektovaný.</p>	Závazný	1
JVK_2.1	JVK – generované systémem – označení Údajů	System umožňuje uživateli označit Údaje v rámci jedné instance objektu Výkaz, pro které nechce vygenerovat JVK (viz kapitola <a href="#">5.5.2.1 Věcné kontroly generované automaticky systémem</a> ).	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	pro negenerování JVK			
<b>JVK_2.2</b>	JVK generované systémem stanovení odchylky JVK systémem	Systém pro JVK vytvořené podle JVK_2.0 stanovuje odchylku na základě pravidel popsaných v kapitole <a href="#">3.20.6 Odchylka v sémantických kontrolách</a> .	Závazný	1
<b>JVK_2.3</b>	JVK generované systémem atributy nastavené systémem	Systém nastavuje JVK vytvořeným podle JVK_2.0 následující atributy: <ul style="list-style-type: none"> <li>• interní identifikátor objektu,</li> <li>• kód objektu,</li> <li>• název objektu,</li> <li>• autor objektu (přihlášený uživatel),</li> <li>• datum vytvoření (aktuální datum),</li> <li>• kdo aktualizoval (přihlášený uživatel),</li> <li>• datum a čas aktualizace (aktuální datum),</li> <li>• platnost_od (platnost_od výkazu),</li> <li>• platnost_do (platnost_do výkazu),</li> <li>• úroveň závažnosti (závažná chyba),</li> <li>• sémantický tvar vzorce kontroly (viz JVK_2.0),</li> <li>• uživatelský tvar kontroly (viz JVK_10.1).</li> </ul>	Závazný	1
<b>JVK_2.4</b>	JVK generované systémem atributy zadávané uživatelem	Systém umožňuje uživateli vyplnit atributy: <ul style="list-style-type: none"> <li>• popis objektu,</li> <li>• poznámka.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .		
<b>JVK_2.5</b>	JVK generované systémem – atributy (zadané systémem) měněné uživatelem	<p>Systém umožňuje uživateli měnit následující atributy JVK (viz kapitola <a href="#">2.3.6 Rámcové vymezení závislosti objektů</a>) podle JVK_2.3:</p> <ul style="list-style-type: none"> <li>• název objektu,</li> <li>• úroveň závažnosti.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	Závazný	1
<b>JVK_2.6</b>	JVK generované systémem – změna odchylky uživatelem	Systém umožňuje uživateli změnit odchylku JVK stanovenou systémem podle JVK_2.2 za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .	Závazný	1
<b>JVK_3.0</b>	JVK – vytvořena uživatelem	<p>Systém umožňuje uživateli ručně vytvořit JVK, která je popsána základními atributy (viz kapitola <a href="#">3.20.1 Atributy objektu Kontrola</a>).</p> <p>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 <a href="#">5.5.2.2 Věcné kontroly vytvářené ručně uživatelem</a>).</p>	Závazný	1
<b>JVK_3.1</b>	JVK – vytvořena uživatelem – sémantický zápis	Systém umožňuje uživateli vytvořit JVK podle JVK_3.0 přímým zápisem sémantického tvaru kontroly.	Závazný	1
<b>JVK_3.2</b>	JVK – vytvořena uživatelem – zápis průvodcem	Systém umožňuje uživateli vytvořit JVK pomocí průvodce (viz kapitola <a href="#">5.5.2.2 Věcné kontroly vytvářené ručně uživatelem</a> ).	Závazný	1
<b>JVK_3.3</b>	JVK – vytvořena	Systém umožňuje uživateli stanovit odchylku JVK přímým zápisem do	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	uživatelé – stanovení odchylky JVK	sémantického tvaru (viz JVK_3.1) nebo pomocí průvodce (viz JVK_3.2).		
<b>JVK_3.4</b>	JVK – vytvořena uživatelem – atributy nastavené systémem	<p>Systém nastavuje JVK vytvořeným podle JVK_3.0 následující atributy:</p> <ul style="list-style-type: none"> <li>• interní identifikátor objektu,</li> <li>• kód objektu,</li> <li>• autor objektu (přihlášený uživatel),</li> <li>• datum vytvoření (aktuální datum),</li> <li>• kdo aktualizoval (přihlášený uživatel),</li> <li>• datum a čas aktualizace (aktuální datum),</li> <li>• platnost_od (platnost_od výkazu),</li> <li>• platnost_do (platnost_do výkazu),</li> <li>• úroveň závažnosti (závažná chyba),</li> <li>• uživatelský tvar kontroly (viz JVK_10.1).</li> </ul>	Závazný	1
<b>JVK_3.5</b>	JVK – vytvořena uživatelem – atributy zadávané uživatelem	<p>Systém umožňuje uživateli vyplnit atributy:</p> <ul style="list-style-type: none"> <li>• název objektu,</li> <li>• popis objektu,</li> <li>• poznámka,</li> <li>• sémantický tvar vzorce kontroly.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	Závazný	1
<b>JVK_3.6</b>	JVK – vytvořena uživatelem – atributy měněné uživatelem	<p>Systém umožňuje uživateli měnit následující atributy JVK podle JVK_3.4:</p> <ul style="list-style-type: none"> <li>• kód objektu,</li> <li>• úroveň závažnosti.</li> </ul>	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a> .		
<b>JVK_4.0</b>	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
<b>JVK_5.0</b>	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 <a href="#">2.3 Vazby mezi jednotlivými objekty</a> ).	Závazný	1
<b>JVK_5.1</b>	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 <a href="#">2.3 Vazby mezi jednotlivými objekty</a> ), pokud byla v systému provedena změna mající vliv pouze na uživatelský tvar JVK.	Závazný	1
<b>JVK_5.2</b>	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
<b>JVK_6.0</b>	JVK – import JVK z externího prostředí	Systém umožňuje importovat kontrolu JVK z formuláře (viz kapitola <a href="#">6.1 Proces Přebírání metapopisu z externích zdrojů</a> ).	Vítaný	3
<b>JVK_7.0</b>	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	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		Projektovaný.		
<b>JVK_8.0</b>	JVK – ukončení platnosti uživatelem	Systém umožňuje uživateli ukončit platnost instanci objektu JVK (viz OBE_11.0).	Závazný	1
<b>JVK_8.1</b>	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
<b>JVK_8.2</b>	JVK – prodloužení platnosti	Systém prodlužuje platnost instance objektu JVK v rámci prodloužení platnosti nadřazené instance objektu Výkaz (viz VYK_5.2).	Závazný	1
<b>JVK_8.3</b>	JVK – smazání	Systém umožňuje uživateli smazat instanci objektu JVK (viz OBE_10.0)	Závazný	1
<b>JVK_9.0</b>	JVK - operátory	Systém disponuje pro tvorbu JVK operátory uvedenými v kapitole <a href="#">3.20.3.1 Kontroly vytvořené sémantickým jazykem</a> ).	Závazný	1
<b>JVK_9.1</b>	JVK – import algoritmu JVK	Systém umožňuje uživateli importovat algoritmus pro JVK (viz kapitola <a href="#">5.5.2.3 Věcné kontroly zapsané algoritmem</a> ).	Závazný	3
<b>JVK_9.2</b>	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
<b>JVK_9.3</b>	JVK – export algoritmu do textu	Systém umožňuje uživateli exportovat algoritmus JVK do textového souboru.	Závazný	3

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>JVK_10.0</b>	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
<b>JVK_10.1</b>	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
<b>JVK_10.2</b>	JVK – nezadaná odchylka	Systém informuje uživatele na chybějící odchylku JVK v sémantickém tvaru JVK.	Závazný	1
<b>JVK_11.0</b>	JVK – text chybového hlášení	Systém na základě definovaného algoritmu (viz dokument <a href="#">D – Sběr dat kapitola Protokol o dokončení zpracování Vstupní zprávy</a> ) vytváří text chybového hlášení, které se odesílá v rámci Výstupní zprávy (viz dokument <a href="#">D – Sběr dat, objekt Výstupní zpráva</a> ).	Závazný	1
<b>JVK_12.0</b>	JVK – výměna Parametru v sémantickém tvaru	Systém umožňuje uživateli provést hromadnou výměnu Parametru v sémantickém tvaru vybraných JVK.	Závazný	3
<b>JVK_13.0</b>	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

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>JVK_13.1</b>	JVK – zobrazení seznamu podle Datové oblasti	System umožňuje uživateli zobrazit seznam všech instancí objektu JVK v rámci jedné instance objektu Datová oblast ve formě tabulky (grid).	Závazný	1
<b>JVK_13.2</b>	JVK – zobrazení zápisu Jednovýkazové kontroly	System 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
<b>JVK_14.0</b>	JVK – prezentace Osobám	System prezentuje JVK Osobám v rámci prezentace instance objektu Výkaz (viz VYK_8.1).	Závazný	2

### 7.17 Mezivýkazová kontrola (MVK)

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>MVK_21.0</b>	MVK – sémantický jazyk MVK	System umožňuje uživateli zapsat MVK prostřednictvím sémantického jazyka pro tvorbu kontrol (viz kapitola <a href="#">3.20.2.1 Kontroly vytvořené sémantickým jazykem</a> a kapitola <a href="#">3.20.5.1 Kontroly vytvořené sémantickým jazykem</a> ).	Závazný	1
<b>MVK_21.1</b>	MVK – algoritmická	System umožňuje uživateli vytvořit algoritmickou MVK (viz kapitola <a href="#">3.20.5.2 Algoritmické kontroly</a> ) výběrem algoritmu pro MVK ze seznamu (viz MVK_29.2) a jeho aplikací na konkrétní Údaje.	Závazný	2
<b>MVK_21.2</b>	MVK – sledování historie	System u každé instance objektu MVK sleduje její historii podle kapitoly <a href="#">3.20.5 Objekt Mezivýkazová kontrola (MVK)</a> .	Závazný	1
<b>MVK_21.3</b>	MVK – vytvoření algoritmu	System je dodán včetně algoritmů pro MVK uvedené v <a href="#">8.1 Příloha 1 – Seznam funkcí pro algoritmické kontroly dodaných se systémem SDAT</a> .	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	dodavatelem			
MVK_22.0	MVK – Skupina MVK	Systém umožňuje uživateli vytvořit Skupinu MVK podle pravidel uvedených v kapitole <a href="#">3.20.5 Objekt Mezivýkazová kontrola (MVK)</a> .	Závazný	1
MVK_23.0	MVK – vytvořena uživatelem	Systém umožňuje uživateli ručně vytvořit MVK, která je popsána základními atributy (viz kapitola <a href="#">3.20.1 Atributy objektu Kontrola</a> ).  Systém umožňuje uživateli vytvořit MVK v sémantickém jazyce (viz MVK_21.0) i algoritmickým zápisem (viz MVK_21.1) a to pouze pro instanci objektu Výkaz, jenž se nachází ve stavu Projektovaný (viz kapitola <a href="#">5.5.2.2 Věcné kontroly vytvářené ručně uživatelem</a> ).	Závazný	1
MVK_23.1	MVK – sémantický zápis	Systém umožňuje uživateli vytvořit MVK podle MVK_23.0 přímým zápisem sémantického tvaru kontroly. Před uložením vzorce systém: <ul style="list-style-type: none"> <li>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 našel,</li> <li>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,</li> <li>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.</li> </ul> Systém umožní 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,	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		upozorní na tuto skutečnost uživatele a nechá ho vybrat, zda si přeje chybu ihned opravit (v takovém případě systém neprovede uložení vzorce MVK, ale umožní 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.		
MVK_23.2	MVK – zápis průvodcem	<p>Systém umožňuje uživateli vytvořit MVK pomocí průvodce (viz kapitola <a href="#">5.5.2.2 Věcné kontroly vytvářené ručně uživatelem</a>). Tento průvodce 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.</p> <p>Systém neumožní uživateli vybírat Hodnoty údaje z jiných Výkazů, než z těch, které jsou definované jako členové dané skupiny MVK.</p>	Závazný	1
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	<p>Systém nastavuje MVK vytvořeným podle MVK_23.0 následující atributy:</p> <ul style="list-style-type: none"> <li>• interní identifikátor objektu,</li> <li>• kód objektu,</li> <li>• autor objektu (přihlášený uživatel),</li> <li>• datum vytvoření (aktuální datum),</li> <li>• kdo aktualizoval (přihlášený uživatel),</li> <li>• datum a čas aktualizace (aktuální datum),</li> <li>• platnost_od (platnost_od výkazu, který je vlastníkem MVK),</li> </ul>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>• platnost_do (platnost_do výkazu, který je vlastníkem MVK ),</li> <li>• úroveň závažnosti (závažná chyba),</li> <li>• uživatelský tvar kontroly (viz MVK_30.0).</li> </ul>		
MVK_23.5	MVK – atributy zadávané uživatelem	<p>Systém umožňuje uživateli vyplnit atributy:</p> <ul style="list-style-type: none"> <li>• název objektu,</li> <li>• popis objektu,</li> <li>• poznámka,</li> <li>• sémantický tvar vzorce kontroly.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	Závazný	1
MVK_23.6	MVK – atributy měněné uživatelem	<p>Systém umožňuje uživateli měnit následující atributy MVK (viz kapitola <a href="#">2.3.6 Rámcové vymezení závislosti objektů</a>) podle MVK_23.4:</p> <ul style="list-style-type: none"> <li>• kód objektu,</li> <li>• úroveň závažnosti.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	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	Systém umožňuje uživateli vytvořit verzi/variantu jakékoliv instanci objektu MVK, jenž je v instanci objektu Výkaz (označený jako Vlastník	Závazný	1



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	uživatel	<p>MVK), který je ve stavu Projektovaný (viz kapitola <a href="#">2.3 Vazby mezi jednotlivými objekty</a>).</p> <p>Úprava instance objektu MVK je možná pouze v instanci objektu Výkaz, který je označen jako vlastník MVK.</p> <p>Systém umožňuje editovat instanci objektu MVK, pokud nadřizená 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.</p> <p>V případě, že nadřizená 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.</p> <p>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.</p>		
<b>MVK_25.1</b>	MVK – vytvoření verze/varianty systémem	<p>Systém vytváří novou verzi/variantu MVK, pouze pokud je v instanci objektu Výkaz označený jako vlastník MVK, který je ve stavu Projektovaný.</p> <p>Systém vytváří novou verzi/variantu instance objektu MVK (viz kapitola <a href="#">2.3 Vazby mezi jednotlivými objekty</a>), pokud byla v systému provedena změna mající vliv pouze na uživatelský tvar MVK.</p> <p>Úprava instance objektu MVK je možná pouze v instanci objektu Výkaz, který je označen jako vlastník MVK.</p>	Závazný	1
<b>MVK_25.2</b>	MVK – změna	Systém umožňuje uživateli změnit vytvořenou variantu instance objektu	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	varianty na verzi	MVK na verzi instance objektu MVK (viz OBE_12.0).		
<b>MVK_26.0</b>	MVK – import MVK z externího prostředí	Systém umožňuje importovat MVK z formuláře (viz kapitola <a href="#">6.1 Proces Přebírání metapopisu z externích zdrojů</a> ).	Vítaný	3
<b>MVK_27.0</b>	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
<b>MVK_28.0</b>	MVK – ukončení platnosti uživatelem v závislosti na zplnění jiné verze/varianty téhož objektu	<p>Systém umožňuje uživateli ukončit platnost instanci objektu MVK (viz OBE_11.0). Ukončení platnosti MVK má tyto následky:</p> <ul style="list-style-type: none"> <li>• pokud nadřazená 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 umožní 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),</li> <li>• pokud nadřazená 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.</li> </ul> <p>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</p>	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		souvisejících instancí objektu Výkaz i samotné instance objektu Výkaz.		
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 nadřazené 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 <a href="#">3.20.5.1 Kontroly vytvořené sémantickým jazykem</a> .	Závazný	1
MVK_29.1	MVK – import algoritmu MVK	Systém umožňuje uživateli importovat algoritmus pro MVK (viz kapitola <a href="#">5.5.2.3 Věcné kontroly zapsané algoritmem</a> ).	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	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		verzi/variantu MVK (viz MVK_25.1) a vytváří této MVK nový uživatelský tvar.		
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 <a href="#">D – Sběr dat kapitola Protokol o dokončení zpracování Vstupní zprávy</a> ) vytváří text chybového hlášení, které se odesílá v rámci Výstupní zprávy (viz dokument <a href="#">D – Sběr dat, kapitola 2.5 Objekt Výstupní zpráva</a> )	Závazný	1
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	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,	Závazný	1

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	kontroly	ve struktuře Datových oblastí instancí objektu Výkaz, které jsou účastníky MVK.		
<b>MVK_34.0</b>	MVK prezentace Osobám	– Systém prezentuje MVK Osobám v rámci prezentace instance objektu Výkaz (viz VYK_8.1).	Závazný	2

### 7.18 Kontrola časové řady (KČŘ)

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>KČŘ_1.0</b>	KČŘ – algoritmická	– Systém umožňuje uživateli vytvořit KČŘ (viz kapitola <a href="#">3.20.4 Objekt Kontrola časové řady (KČŘ)</a> ) výběrem algoritmu KČŘ ze seznamu (viz KČŘ_6.1) a jeho aplikací na konkrétní Údaj nebo Údaje.	Závazný	2
<b>KČŘ_1.1</b>	KČŘ – sledování historie	Systém u každé instance objektu KČŘ sleduje její historii podle kapitoly <a href="#">3.20.4 Objekt Kontrola časové řady (KČŘ)</a> .	Závazný	2
<b>KČŘ_1.2</b>	KČŘ – vytvoření algoritmu dodavatelem	Systém je dodán včetně algoritmu pro KČŘ uvedeného v příloze <a href="#">8.1 Příloha 1 – Seznam funkcí pro algoritmické kontroly dodaných se systémem SDAT</a> .	Závazný	2
<b>KČŘ_2.0</b>	KČŘ – atributy nastavené systémem	Systém nastavuje KČŘ vytvořeným podle KČŘ_1.0 následující atributy: <ul style="list-style-type: none"> <li>• interní identifikátor objektu,</li> <li>• kód objektu,</li> <li>• název objektu,</li> <li>• autor objektu (přihlášený uživatel),</li> <li>• datum vytvoření (aktuální datum),</li> <li>• kdo aktualizoval (přihlášený uživatel),</li> </ul>	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		<ul style="list-style-type: none"> <li>datum a čas aktualizace (aktuální datum),</li> <li>platnost_od (platnost_od výkazu),</li> <li>platnost_do (platnost_do výkazu),</li> <li>úroveň závažnosti (závažná chyba, pro kontrolu lineární regrese chyba k potvrzení).</li> </ul>		
KČŘ_2.1	KČŘ – atributy zadávané uživatelem	<p>Systém umožňuje uživateli vyplnit atributy:</p> <ul style="list-style-type: none"> <li>popis objektu,</li> <li>poznámka.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	Závazný	2
KČŘ_2.2	KČŘ – atributy (zadané systémem) měněné uživatelem	<p>Systém umožňuje uživateli měnit následující atributy KČŘ (viz kapitola <a href="#">2.3.6 Rámcové vymezení závislosti objektů</a>) podle KČŘ_2.0:</p> <ul style="list-style-type: none"> <li>kód objektu,</li> <li>název objektu,</li> <li>úroveň závažnosti.</li> </ul> <p>Tyto atributy umožňuje systém uživateli měnit za dodržení pravidel stanovených v kapitole <a href="#">2.3 Vazby mezi jednotlivými objekty</a>.</p>	Závazný	2
KČŘ_3.0	KČŘ – vytvoření verze/varianty uživatelem	<p>Systém umožňuje uživateli vytvořit verzi/variantu jakékoliv instanci objektu KČŘ, jenž je v instanci objektu Výkaz, který je ve stavu Projektovaný (viz kapitola <a href="#">2.3 Vazby mezi jednotlivými objekty</a>).</p>	Závazný	2
KČŘ_3.1	KČŘ – změna varianty na verzi	<p>Systém umožňuje uživateli změnit vytvořenou variantu instance objektu KČŘ na verzi instance objektu KČŘ (viz OBE_12.0).</p>	Závazný	2
KČŘ_4.0	KČŘ – unikátnost atributů	<p>Systém zajišťuje unikátnost atributů kód objektu a název instance objektu KČŘ v rámci jedné verze/varianty Výkazu. Nepovolí uživateli založit</p>	Závazný	2

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
		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	System umožňuje uživateli ukončit platnost instanci objektu KČŘ (viz OBE_11.0).	Závazný	2
KČŘ_5.1	KČŘ – ukončení platnosti systémem v závislosti na zplnění jiné verze/varianty téhož objektu	System automaticky ukončí platnost instanci objektu KČŘ (viz OBE_11.1).	Závazný	2
KČŘ_5.2	KČŘ – prodloužení platnosti	System prodlužuje platnost instance objektu KČŘ v rámci prodloužení platnosti nadřazené instance objektu Výkaz (viz VYK_5.2).	Závazný	2
KČŘ_5.3	KČŘ – smazání	System umožňuje uživateli smazat instanci objektu KČŘ (viz OBE_10.0).	Závazný	2
KČŘ_6.0	KČŘ – import algoritmu KČŘ	System umožňuje uživateli importovat algoritmus pro KČŘ v souladu s kapitolou <a href="#">5.5.2.3 Věcné kontroly zapsané algoritmem</a> .	Závazný	3
KČŘ_6.1	KČŘ – seznam algoritmů KČŘ	System 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 algoritmu do	System umožňuje uživateli exportovat algoritmus KČŘ do textového souboru.	Závazný	2



ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
	textu			
KČŘ_7.0	KČŘ – text chybového hlášení	Systém na základě definovaného algoritmu (viz dokument <a href="#">D – Sběr dat, kapitola Protokol o dokončení zpracování Vstupní zprávy</a> ) 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 <a href="#">5.5.2.3 Věcné kontroly zapsané algoritmem</a> ) 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 <a href="#">3.20.4.1 Objekt Mezisubjektová kontrola (MSK) jako specifický typ KČŘ</a> ).	Závazný	2

## 7.19 Knihovna

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
<b>KNH_1.0</b>	Zobrazení objektů Knihovny	Systém umožňuje uživateli zobrazit náhled na instance objektů Knihovny (viz kapitola <a href="#">4.1 Knihovna</a> ).	Závazný	1
<b>KNH_2.0</b>	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
<b>KNH_2.1</b>	Uživatelské nastavení navigace	Systém umožňuje uživateli nastavit „svůj“ default navigace v Knihovně.	Závazný	2
<b>KNH_3.0</b>	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.		
<b>KNH_4.0</b>	Historie atributů objektů Knihovny	Systém umožňuje uživateli zobrazit historii jednotlivých atributů a vazeb.	Závazný	1
<b>KNH_5.0</b>	Použití objektů Knihovny	<p>Systém umožňuje uživateli zobrazit použití jednotlivých instancí objektů v nadřazených objektech:</p> <ul style="list-style-type: none"> <li>• v časovém řezu objektu,</li> <li>• do budoucna,</li> </ul> <p>podle kapitoly <a href="#">2.4.1 Základní vlastnosti objektů</a>.</p> <p>Systém zobrazuje použití objektu v tabulce (grid). Uživatel může provádět filtry a zobrazit detail vybraného objektu.</p>	Závazný	1
<b>KNH_6.0</b>	Nepoužité objekty Knihovny	<p>Systém umožňuje uživateli zobrazit instance objektů, které nemají použití v nadřazených objektech:</p> <ul style="list-style-type: none"> <li>• v časovém řezu objektu,</li> <li>• do budoucna.</li> </ul> <p>Systém zobrazuje nepoužité objekty v tabulce (grid). Uživatel může provádět filtry a zobrazit detail vybraného objektu.</p>	Závazný	1

## 7.20 Navazování časových řad

ID požadavku	Název požadavku	Popis požadavku	Důležitost	Kategorie
CAR_1.0	Navazování časové řady Údaje	System umožňuje uživateli navázat do časové řady Údaje dle kapitoly <a href="#">6.2 Proces Navazování časových řad Údajů</a> .	Závazný	3
CAR_2.0	Zajištění časových řad při nezměněném popisu Údaje	System automaticky zajišťuje časové řady Údajů, které v jednotlivých instancích Výkazu nezměnily popis.	Závazný	3
CAR_3.0	Kontrola navázání časové řady	System umožňuje uživateli spustit kontrolu správnosti navázání časové řady Údajů (viz kapitola <a href="#">6.2.2 Průběh procesu</a> ).	Závazný	3
CAR_4.0	Zobrazení přerušovaných časových řad Údajů	System umožňuje uživateli zobrazit přerušené časové řady Údajů v grafickém vzoru Výkazu.	Závazný	3
CAR_4.1	Zobrazení hloubek časových řad Údajů	System 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	System umožňuje uživateli zobrazit řetězec časové řady Údaje vč. funkčního výrazu.	Závazný	3
CAR_5.0	Rozvázání časové řady	System umožňuje uživateli rozpojit jednou navázané časové řady Údajů.	Závazný	3

## 8 Přílohy

### 8.1 Příloha 1 — Seznam funkcí pro algoritmické kontroly dodaných se systémem 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 zaslánymi organizátorem regulovaného trhu, tzn. Burzou cenných papírů Praha (dále jen BCPP), a výkazy zaslánymi 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 vykázané 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.

## 8.2 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 nadřazených uzlových Položek hierarchie vložené Položky hierarchie v dané Hierarchii číselníku. Tato změna se musí promítnout do všech souvisejících Hierarchií 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 <b>E5</b>	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	K1 E1 E2 U2 U21	Hierarchie2 obsahuje společnou uzlovou položku U2 a společnou kořenovou položku K1 s Hierarchií1 a bude synchronizována.  Položka hierarchie E5 je vložena do Hierarchie2 na

E3 E4	E3 E4  <b>E5</b>	<p>nejnižší možnou úroveň společné uzlové položky U2 tj. na úroveň Položky hierarchie U21.</p> <p>Položka hierarchie E5 se stává elementární položkou pro uzlovou položku U2 a kořenovou položku K1.</p>
<b>Hierarchie3 – původní stav</b>	<b>Hierarchie3 – po synchronizaci</b>	<b>Popis synchronizace</b>
<p>K2</p> <p style="padding-left: 40px;">K1</p> <p style="padding-left: 80px;">E1</p> <p style="padding-left: 80px;">E2</p> <p style="padding-left: 80px;">E3</p> <p style="padding-left: 80px;">E4</p> <p style="padding-left: 40px;">U3</p> <p style="padding-left: 80px;">E6</p> <p style="padding-left: 80px;">E7</p>	<p>K2</p> <p style="padding-left: 40px;">K1</p> <p style="padding-left: 80px;">E1</p> <p style="padding-left: 80px;">E2</p> <p style="padding-left: 80px;">E3</p> <p style="padding-left: 80px;">E4</p> <p style="padding-left: 80px;"><b>E5</b></p> <p style="padding-left: 40px;">U3</p> <p style="padding-left: 80px;">E6</p> <p style="padding-left: 80px;">E7</p>	<p>Hierarchie3 obsahuje kořenovou položku K2, jež má podřízenou uzlovou položku K1.</p> <p>Položka hierarchie E5 je při synchronizaci vložena pod společnou uzlovou položku K1 s Hierarchií1.</p> <p>Položka hierarchie E5 se stává elementární položkou uzlové položky K1 a kořenovou položku K2.</p>

Tabulka 7 - 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 nadřazených 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:

Hierarchie 1 – původní stav	Hierarchie 1 – 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 E1 E2 U2 E3 E4 E5 U3 E6 E7	K2 E1 E2 U2 E3 E4 E5 U3 E6 E7	Hierarchie2 obsahuje společnou 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	K2 K1 E1	Hierarchie3 obsahuje společnou uzlovou položky K1 s Hierarchií1. Položka E5 je přímo pod K1, a protože uzel

	E2		E2	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 nadřazené uzlové položky K2. Uzlové položka U3 zůstává nezměněna.
	E3		E3	
	E4		E4	
	E5		<del>E5</del>	
U3		U3		
	E6		E6	
	E7		E7	

Tabulka 8 - 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	K1	<p>Při aktualizaci Hierarchie1 byla smazána uzlová položka U2 z kořene hierarchie K1.</p> <p>Elementární položky E3,E4,E5 zůstaly v Hierarchii1 a jejich přímou nadřazenou se stala kořenová položka K1, jinak nedošlo ke změně elementárního obsahu žádné uzlové položky.</p> <p>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</p>
U1	U1	
E1	E1	
E2	E2	
U2	<del>U2</del>	
E3	E3	
E4	E4	
E5	E5	

Hierarchie2 – původní stav	Hierarchie2 – po synchronizaci	Popis synchronizace
K2	K2	ovlivněn.
E1	E1	Beze změny. Uzlová položka U2 je zde metodiky požadována.
E2	E2	
U2	U2	
E3	E3	
E4	E4	
E5	E5	
U3	U3	
E6	E6	
E7	E7	

Tabulka 9 - 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	K1	Při aktualizaci Hierarchie1 byla smazána uzlová položka U2 včetně svého elementárního obsahu.
U1	U1	
E1	E1	
E2	E2	
U2	<del>U2</del>	
E3	<del>E3</del>	Elementární obsah uzlové položky U2 se nezměnil, změnil se ovšem elementární obsah kořenové položky K1.
E4	<del>E4</del>	Dále se uplatňuje pravidlo 2.
E5	<del>E5</del>	
Hierarchie2 – původní stav	Hierarchie2 – po aktualizaci	Popis synchronizace

stav	synchronizaci	
K2	K2	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 nadřizená kořenová položka K2. Elementární Obsah uzlové položky U3 zůstane nezměněn.
K1	K1	
E1	E1	
E2	E2	
<del>E3</del>	<del>E3</del>	
<del>E4</del>	<del>E4</del>	
<del>E5</del>	<del>E5</del>	
U3	U3	
E6	E6	
E7	E7	

Tabulka 10 - 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	<b>K1</b>	Při aktualizaci Hierarchie1 byla vložena uzlová položka U3 pod kořenovou položku K1 a nad uzlové položky U1 a U2.
U1	<b>U3</b>	
E1	U1	V rámci synchronizace se musí provést kontrola, zda nedošlo k záměně uzlů.
E2	E1	
U2	E2	Položka U3 nebyla použita v žádném jiné Hierarchii číselníku (byla elementární položkou Číselníku). Po vložení do Hierarchie1 se změnil se její
E3	U2	
E4	E3	
E5	E4	
	E5	

		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“.
Hierarchie 2 – původní stav	Hierarchie 2 – 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 11 - Vložení uzlové položky do Hierarchie číselníku (Příklad 1)

Příklad 2 – ukázka konfliktu a nepovolené úpravy Hierarchie číselníku

Hierarchie1 – původní stav	Hierarchie 1 – po aktualizaci	Popis aktualizace
K1  U1 E1 E2 U2 E3 E4 E5	K1  <b>U3</b> 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  <b>U3 KONFLIKT</b>  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.  <b>Výsledek – synchronizace není povolena a končí chybou.</b>
--	--	--

Tabulka 12 - 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  <b>U3</b>  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	K2  U3  E1 E2 E3 E4	Při synchronizaci dojde ke smazání elementární položky E6 z uzlové položky U3 v Hierarchii2.  Změna elementárního obsahu promítne do kořenové položky K2.

E5	E5	
E6	<del>E6</del>	

Tabulka 13 - 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 E4 E5 E6	K1 U1 E1 E2 U2 <del>E3</del> <del>E4</del> <del>E5</del> <del>E6</del>	<p>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.</p> <p>Protože položka U2 je pod kořenovou položkou K1, byly elementární položky E3, E4, E5, E6 smazány i z K1.</p> <p>Výstupem aktualizace je zrušení elementárních položek E3, E4, E5, E6 z uzlových položek U2 a K1.</p>
Hierarchie2 – původní stav	Hierarchie2 – po synchronizaci	Popis synchronizace
K1 E1 E2	K1 E1 E2	<p>Hierarchie2 obsahuje uzlové položky U2 i K1 a bude synchronizována.</p> <p>Uzlová položka U2</p>



<p>U2</p> <p>U3</p> <p>E3</p> <p>E4</p> <p>U4</p> <p>E5</p> <p>E6</p>	<p>U2</p> <p><del>U3</del></p> <p><del>E3</del></p> <p><del>E4</del></p> <p><del>U4</del></p> <p><del>E5</del></p> <p><del>E6</del></p>	<p>obsahuje další úroveň uzlových položek U3 (E3, E4) a U4 (E5, E6).</p> <p>Položky E3, E4, E5, E6 budou ze synchronizované Hierarchie číselníku zrušeny i se svojí nadřazenou úrovní U3 a U4 (viz kapitola <a href="#">8.2.8 Pravidlo 7 — Smazání celého uzle hierarchie včetně uzlové položky</a>).</p> <p>Jelikož byly zrušeny uzlové položky U3 a U4, uplatní se pravidlo 3. Žádný další sekundární efekt zde proto nevzniká.</p>
Hierarchie 3 – původní stav	Hierarchie3 – po synchronizaci	Popis synchronizace
<p>K1</p> <p>E1</p> <p>E2</p> <p>E3</p> <p>E4</p> <p>E5</p> <p>E6</p>	<p>K1</p> <p>E1</p> <p>E2</p> <p><del>U2</del></p> <p><del>E3</del></p> <p><del>E4</del></p> <p><del>E5</del></p> <p><del>E6</del></p>	<p>Hierarchie 3 obsahuje pouze kořenovou položku K1. Položky E3, E4, E5, E6 jsou přímo pod ní a při synchronizaci se zruší.</p> <p>Naopak se vloží položka U2, která se stala po synchronizaci elementární.</p>

Tabulka 14 - 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  $\Sigma$ .

Hierarchie1 – původní stav	Hierarchie1 – po	Popis aktualizace
----------------------------	------------------	-------------------

	aktualizaci	
K1 U1 E1 E2 U2	K1 U1 E1 E2 U2 <b>E3</b> <b>E4</b> <b>E5</b>	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 nadřizenou položku K1. Byl změněn elementární rozpad položky K1.
Hierarchie2 – původní stav	Hierarchie2 – po synchronizaci	Popis synchronizace
K1 E1 E2 U2	K1 E1 E2 U2 <b>E3</b> <b>E4</b> <b>E5</b>	Hierarchie2 obsahuje jednu společnou uzlovou položku K1 s Hierarchií 1 a bude synchronizována.  Položka U2 se vložním dalších položek o úroveň níže stává položkou uzlovou.

Tabulka 15 - 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 podřizené 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	K1 U1 E1 E2	Cílem aktualizace je zrušení celého uzle U2, tj. uzlové položky U2 včetně jejího elementárního obsahu.

U2 E3 E4 E5	<del>U2</del> <del>E3</del> <del>E4</del> <del>E5</del>	Výstupem je změna elementárních položek kořenové položky K1, obsah uzlové položky U1 zůstává nezměněn.
<b>Hierarchie2 – původní stav</b>	<b>Hierarchie2 – po synchronizaci</b>	<b>Popis synchronizace</b>
K1 E1 E2 U2 E3 E4 E5	K1 E1 E2 <del>U2</del> <del>E3</del> <del>E4</del> <del>E5</del>	Hierarchie2 obsahuje kořenovou položku K1 a 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 16 - 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.
<b>Hierarchie2 – původní stav</b>	<b>Hierarchie2 – po</b>	<b>Popis synchronizace</b>

	synchronizaci		
K2	K2		<p>Hierarchie2 již obsahuje elementární položku E6 pod uzlem U3.</p> <p>Vzniká konflikt a Hierarchie číselníku nelze synchronizovat.</p> <p>Bylo porušeno pravidlo, že jedna položka smí být obsažena v Hierarchii číselníku pouze jedenkrát.</p> <p><u>Řešení:</u> uživatel se musí rozhodnout, zda je požadovaná akce z věcného hlediska správná a upravit Hierarchii číselníku postupně.</p> <p>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.</p>
U1	U1		
E1	E1		
E2	E2		
U2	U2		
E3	E3		
E4	E4		
E5	E5		
U3	<b>E6</b>		
E6	U3		
E7	<b>E6</b>		
	E7		
	<b>KONFLIKT</b>		

Tabulka 17 - 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	K1	V rámci aktualizace byla pod uzlovou položku U1 přidána elementární položka E6.
U1	U1	
E1	E1	
E2	E2	

U2 E3 E4 E5	<b>E6</b> U2 E3 E4 E5	
Hierarchie2 – původní stav	Hierarchie2 – po synchronizaci	Popis synchronizace
K2 U1 E1 E2 U2 E3 E4 E5 E6 E7	K2 U1 E1 E2 <b>E6</b> U2 E3 E4 E5 <b>E6</b> E7 <b>KONFLIKT</b>	<p>Hierarchie2 obsahuje již elementární položku E6 pod uzlem K2.</p> <p>Vzniká konflikt a Hierarchie číselníku nelze synchronizovat.</p> <p>Bylo porušeno pravidlo, že jedna Položka hierarchie smí být obsažena v Hierarchii číselníku pouze jedenkrát.</p>

Tabulka 18 - 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	K1 U1 E1 E2	Do Hierarchie1 byla vložena položka E4

<p style="text-align: center;">E3</p> <p style="text-align: center;">U2</p> <p style="text-align: center;">F1</p> <p style="text-align: center;">F2</p> <p style="text-align: center;">F3</p>	<p style="text-align: center;">E3</p> <p style="text-align: center;"><b>E4</b></p> <p style="text-align: center;">U2</p> <p style="text-align: center;">F1</p> <p style="text-align: center;">F2</p> <p style="text-align: center;">F3</p>	
Doména1 - původní stav	Doména1 – stav po aktualizaci	Popis aktualizace
<p>K1 <span style="float: right;">Σ</span></p> <p style="padding-left: 20px;">U1 <span style="float: right;">Σ</span></p> <p style="padding-left: 40px;">E1</p> <p style="padding-left: 40px;">E2</p> <p style="padding-left: 40px;">E3</p> <p style="padding-left: 20px;">U2 <span style="float: right;">Σ</span></p> <p style="padding-left: 40px;">F1</p> <p style="padding-left: 40px;">F2</p> <p style="padding-left: 40px;">F3</p>	<p>K1 <span style="float: right;">Σ</span></p> <p style="padding-left: 20px;">U1 <span style="float: right;">Σ</span></p> <p style="padding-left: 40px;">E1</p> <p style="padding-left: 40px;">E2</p> <p style="padding-left: 40px;">E3</p> <p style="padding-left: 20px;">U2 <span style="float: right;">Σ</span></p> <p style="padding-left: 40px;">F1</p> <p style="padding-left: 40px;">F2</p> <p style="padding-left: 40px;">F3</p>	<p>Z Hierarchie byla vytvořena ruční Doména1, která kopíruje celou Hierarchii číselníku.</p> <p>Tato Doména číselníku je úplná ve všech uzlových položkách K1, U1, U2. Vložení položky E4 do hierarchie se projeví vložím položky E4 pod položku U1 (viz kapitola <a href="#">3.9 Objekt Doména číselníku bod 8</a>)</p>
Doména2 - původní stav	Doména2 – stav po aktualizaci	Popis aktualizace
<p>K1</p> <p>U1</p> <p>F1</p> <p>F2</p> <p>F3</p>	<p>K1</p> <p>U1</p> <p>F1</p> <p>F2</p> <p>F3</p> <p>Beze změny</p>	<p>Doména2 je vytvořena přímo vybráním položek z číselníku.</p> <p>Změna v hierarchii se domény nedotkne.</p>

Tabulka 19 – Aktualizace Domén číselníku – Pravidlo 1

### 8.3.2 Pravidlo 2

Aktualizace se týká pouze úplných uzlů ( $\Sigma$ ) 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                    Σ U1                Σ E1 E2 E3 U2 F1 F2	K1                    Σ U1                Σ 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 vložím 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 20 - 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 nadřazeného úplného uzlu nezměnil.

Hierarchie- původní stav	Hierarchie -po aktualizaci	Popis aktualizace
K1  U1  E1 E2 E3  U2  F1 F2 F3	K1  U1  E1 E2 E3  <del>U2</del> F1 F2 F3	Z Hierarchie číselníku byla zrušena uzlová položka U2. Elementární položky F1, F2, 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	K1  U1  E1 E2 E3  <del>U2</del> F1	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



F2	F2	číselníku neměnil.
F3	F3	

Tabulka 21 - Aktualizace Domén číslníku – Pravidlo 4

#### 8.4 Příloha 4 — Počty instancí vybraných objektů metapopisu v roce 2015 v systému MtS

Objekt	Počet	Poznámka
Výkaz	178	
Datová oblast	883	
Číselník	210	
Položka číslníku	8010	Bez vykazovacích povinností.
Hierarchie číslníku	336	
Doména číslníku	3670	Včetně automatických domén vzniklých z Hierarchie číslníku a bez domén potřebných pro vykazovací povinnosti.
Převodník	2	
Účtová osnova	2	
Účet	499	
Datový typ	157	
Doména datového typu	58	
Ukazatel	7049	
Parametr	370	Bez vykazovacích povinností a technologických parametrů.
Jednovýkazová kontrola	12910	
Mezivýkazová kontrola	2114	
Skupina MVK	147	
Kontrola v časové řadě	402	Jen odchýlení od lineární regresní přímky.