Jihočeská univerzita v Českých Budějovicích

Přírodovědecká fakulta

Analýza a funkční návrh informačního systému multilingvální diář s hierarchií uživatelů

Bakalářská práce

Autor: Martin Janda

Vedoucí práce: doc. Ing. Ladislav Beránek, CSc., MBA

České Budějovice 2016

Bibliografické údaje

Janda, M., 2016: Analýza a funkční návrh informačního systému multilingvální diář s hierarchií uživatelů [Functional analysis and design of the information system multilingual diary with users hierarchy] – 114 p., Faculty of Science, The university of South Bohemia, České Budějovice, Czech Republic.

Anotace

Tato bakalářská práce se zabývá návrhem informačního systému „multilingvální diář s hierarchií uživatelů“. Návrh je realizován za pomoci nástroje PowerDesigner firmy Sybase. V teoretické části je zahrnuto seznámení se s vývojovým prostředím popis metodik a s analýzou diáře. V praktické části jsou poznatky z teoretické části využity k navrhování konceptuálního, procesního a fyzického modelu. V poslední části práce jsou popsány možnosti vygenerování databáze z fyzického modelu.

Abstract

This bachelor thesis describes the design of information system “multilingual diary hierarchy of users“. The design is implemented with the help of a tool, Sybase PowerDesigner. The theoretical part contains to get to know with the development environment and description of methods of analysis of the diaries. In the practical part this knowledge used for designing conceptual, procedural and physical model. The last part describes the options for generating database from the physical model.

Prohlašuji, že svojí bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

České Budějovice, 22. 04. 2016

Martin Janda

Poděkování

Rád bych poděkoval panu doc. Ing. Ladislavu Beránkovi, CSc., MBA., za cenné rady a připomínky, jimiž přispěl ke zpracování této bakalářské práce. Rovněž bych rád poděkoval společnosti BM servis s.r.o. za zapůjčení softwaru.

Obsah

1 Úvod

Cílem bakalářské práce je zpracovat analýzu a návrh informačního systému „multilingvální diář s hierarchií uživatelů“, který se bude moci využít v podnikové sféře pro lepší plánování času a jednoduššímu sdílení informací. Cílem bude dosáhnout základní funkčnosti, viz kapitola Cíle bakalářské práce s možností budoucího rozšíření. Návrh bude realizován za pomoci nástroje PowerDesigner firmy Sybase, ze kterého se využije technika konceptuálního datového modelu, fyzického datového modelu a procesního modelu.

Práce je rozdělena do těchto částí:

- Na počátku práce bude v teoretickém úvodu vysvětleno, jaké důvody vedly k návrhu IS „multilingvální diář“. Postupně si představíme vývoj IS, zejména datové a procesní modelování. Dále si představíme použitý nástroj PowerDesigner v této bakalářské práci.
- V další kapitole se zaměříme na analýzu požadavků, které jsou zde zpřesněny a detailizovány oproti původnímu strohému znění.
- Návrh řešení je vyjádřen konceptuálním datovým modelem a procesním modelem. Současně je v těchto kapitolách vysvětleno, proč sestavovat modely a jak se modeluje pomocí Sybase PowerDesigner 15.
- V poslední části práce se zaměříme na realizaci řešení. Zde se budeme věnovat fyzickému datovému modelu a jeho odvození z konceptuálního datového modelu. Na závěr se zaměříme na generování skriptu obsahujícího příkazy pro vytvoření struktury databáze, tj. tabulek, jejich sloupců a dalších databázových objektů, v systém řízení báze dat (DBMS) SAP Sybase SQL Anywhere 12.

Součástí této práce jsou elektronické přílohy, které obsahují zmíněné modely a případné další výstupy z modelovacího nástroje PowerDesigner. Tyto přílohy jsou vlastním návrhem budoucího IS „multilingvální diář“.

V kapitolách věnujících se návrhu i realizaci řešení jsem uvedl řadu obrázků reprezentujících diagramy modelů, aby bylo dosaženo větší srozumitelnosti uvedených popisů.

2 Cíle bakalářské práce

Cílem bakalářské práce je zpracovat analýzu a návrh informačního systému „multilingvální diář s hierarchií uživatelů“, který bude zahrnovat:

a) Konceptuální datový model.

b) Fyzický datový model umožňující přímé vygenerování objektů (tabulky, sloupce, indexy, reference, …) databáze Sybase SQL Anywhere.

c) Procesní model definující předpokládané posloupnosti aktivit jednotlivých uživatelů.

Modely budou vytvořeny pomocí nástroje Sybase PowerDesigner 15. Přičemž základní funkční požadavky jsou:

a) Možnost rezervovat čas jednotlivých zdrojů (zejména uživatel) s popisem důvodu rezervace.

b) Signalizace kolizí.

c) Sdílení diářů dle rolí jednotlivých uživatelů (omezení přístupovými právy).

d) Možnost zadávat textové údaje ve více jazycích. Jazyk je jednou z vlastností uživatele.

e) Datové objekty diáře musí být definovány tak, aby umožnily budoucí rozšíření a vazby na datové objekty jiného typu (adresář, zprávy, …), tj. významná míra rozšíření.

3 Teoretický úvod a východiska

Při realizaci této bakalářské práce se uplatnila řada technik a metod systémového inženýrství pro návrh informačních systémů, zejména datové a procesní modelování ve vazbě na analýzu požadavků pomocí CASE nástrojů. Navrhovaný informační systém je po stránce věcné orientován do oblasti time managementu.

V této kapitole se věnuji teoretickým aspektům této práce, jejich zasazení do obecnějšího rámce a použitým nástrojům z obecného pohledu.

3.1 Time management

Time Management neboli řízení času je oblast řízení, která se zaměřuje na hospodaření s časem. Čas je zde považován za zdroj (omezující zdroj v dosahování cílů) podobně jako jiné zdroje, například materiální zdroje. Metody řízení času směřují k získání kontroly nad časem, kterého mají všichni stejně a který jedinci či společnosti tráví při své činnosti, resp. na jednotlivých procesech. Cílem metod řízení času je zvýšení efektivnosti využití času, následně tedy i produktivity v dosahování cílů.[1] Tyto metody se uplatňují nejen v rámci podniků a organizací, ale také v osobním životě.

Řízení času je také nezbytnou součástí projektového řízení, kde se pro zobrazení projektových úkolů v čase (řízení času na projektu) používá:

- Ganttův diagram – sloupce odpovídají kalendáři a řádky strukturovanému rozvrhu prací (WBS – work breakdown structure). [2]
- Síťový diagram – graf, který se skládá z uzlů, které představují jednotlivé činnosti a dobu jejich trvání, propojených hranami, které vyjadřují návaznost činností. [3]

V souvislosti s řízením času na projektu se určuje kritická cesta, tj. nejdéle trvající posloupnost činností mezi zahájením a ukončením projektu. Každé zpoždění na této cestě se projeví, jako celkové zpoždění projektu. Při plánování každé činnosti projektu lze obvykle stanovit, zda má končit co nejdříve (EF – earliest finish) anebo co nejpozději (LF – latest finish). Řízení projektů podporuje řada SW nástrojů (např. Microsoft Project), případně je součástí některých komplexních firemních informačních systémů.

S efektivním plánováním, tedy i využíváním času souvisí metody Time Managementu zaměřující se na uspořádání úkolů směřujících k dosažení cílů dle určitých pravidel, na jejich rozsah a splnitelnost a na přístup k vyhodnocení plnění úkolů. S disciplínou řízení času souvisí i techniky pro stanovení priorit cílů či úkolů, například ABC analýza a Paretova analýza. [1]

ABC analýza se zaměřuje na rozdělení úkolů do skupin, jejich kategorizaci. Pokud se jedná o rozdělení do tří skupin, což je nejobvyklejší, bývají označeny písmeny A, B a C (odtud její název). Kritéria pro rozdělení mohou být tato:

- Úkoly po stránce významu důležité a z hlediska termínu naléhavé.
- Úkoly po stránce významu důležité a hlediska termínu nenaléhavé.
- Úkoly po stránce významu nedůležité a hlediska termínu nenaléhavé.

Paretova analýza vychází z toho, že 80 % úkolů může být obvykle hotovo za 20 % disponibilního času, zatímco zbývajících 20 % úkolů si vezme 80 % času. Tedy by se úkolům z první skupiny měla přiřadit vyšší priorita. [1]

Valná většina metod řízení času tedy v sobě integruje jeho plánování mnohdy s vazbou na cíle (jejich význam a termín) a se zpětnou vazbou pro vyhodnocení skutečnosti oproti plánu. Je zřejmé, že zejména v týmovém internacionálním pojetí je nezbytné plánování času a hospodaření s ním podpořit informačním systémem, k čemuž by měla přispět i tato práce.

3.2 Vývoj informačních systémů

Na veškeré činnosti, které souvisí s vývojem a správou informačních systému, lze nahlížet jako na disciplíny softwarového inženýrství.

Analýza požadavků na IS a navazující návrh IS patří mezi nejdůležitější fáze v životním cyklu IS. Kvalitní návrh vede k úspěšné realizaci (vytvoření datových struktur, programování, instalace, počáteční nastavení a betatestování) a k úspěšnému následnému provozování IS. Nedostatky či dokonce chyby v návrhu IS se nepříznivě projeví v závěrečných fázích vývoje, kdy mohou projekty vývoje systému významně zpozdit a prodražit (ba i způsobit úplný kolaps těchto projektů), ale zejména při budoucím rozvoji IS, kde mohou způsobit situaci, že místo rozvoje stávajícího IS je nutné IS vybudovat nově. [7]

Vzhledem k významu kvalitního návrhu IS byla vyvinuta celá řada metod a metodik. Valná většina z nich se opírá o datové modelování (konceptuální datový model), který specifikuje informační schopnost budoucího IS. [4] Také základem této práce je návrh konceptuálního datového modelu.

Jiné modely se používají k identifikaci, dekompozici a popisu firemních procesů jako další součásti návrhu IS. Obdobně lze užít tyto modely na specifikaci pracovních postupů při ovládání IS. Tyto modely jsou také součástí této práce.

Uvedené modely představující návrh IS na konceptuální úrovni jsou během procesu jejich formování trvale revidovány oproti požadavkům, které jsou touto analýzou zase zpětně zpřesňovány. Účast zadavatele (autora požadavků) je tedy v této fázi návrhu IS nezbytná – vysvětluje „svůj“ reálný svět reprezentovaný požadavky, které se zpřesňují, a odsouhlasuje analytický pohled na něj vyjádřený modely. Výsledkem takovéto realizace návrhu IS nejsou jen modely, ale také úplné vzájemné vyjasnění požadavků a očekávání od budoucího IS mezi zadavatelem a vývojářem. [4]

Druhá, stejně důležitá, role uvedených modelů s návrhem IS spočívá v tom, že je lze použít jako zadání programátorům, jaký IS mají zrealizovat neboli naprogramovat. Součástí této práce je také „malé“ nahlédnutí do realizace IS, a to specifikací fyzického datového modelu a sestavením skriptu pro vygenerování databáze a objektů databáze.

Vzhledem k důležitosti a významu návrhu, ale i správy celého životního cyklu, IS byly vyvinuty a uvedeny do praxe softwarové nástroje, které tyto činnosti podporují. Obvykle jsou označovány zkratkou CASE, kterou lze interpretovat „Computer Aided Systems/Software Engineering“. Některé mají komplexní charakter, jiné se zaměřují jen na určitou fázi životního cyklu, také je lze rozdělit dle možností uživatelského přizpůsobení konkrétním podmínkám a zvyklostem v té či oné vývojářské dílně. [4]

V následujících kapitolách se ještě zaměříme na obecné principy datové modelování a procesního modelování, které se uplatňuje v této práci.

3.2.1 Datové modelování

V rámci datového modelování se do diagramů uspořádávají klíčové objekty modelu (entity nebo tabulky a pohledy databáze) se specifikací jejich vztahů (relace nebo reference) a jejich vlastností (atributy nebo sloupce tabulek). Diagramy se prezentují v přehledné, dobře srozumitelné, grafické formě, která je daná použitou notací. Cílem modelů je usnadnit orientaci v modelované realitě, tedy v reálném světě či v datové struktuře IS. [4]

Datové modelování lze rozdělit dle určení cílového modelu takto:

- Ideální datový model (IDM) – modeluje požadavky, tj. specifikuje požadovanou informační schopnost budoucího IS.
- Konceptuální datový model (CDM) – pomáhá analyzovat konceptuální strukturu navrhovaného IS. Zaměřuje se na identifikaci klíčových typových objektů reálného světa (entity), na identifikaci jejich klíčových vlastností i na identifikaci jejich vztahů. Na identifikaci navazuje návrh, tj. takové uspořádání entit, které koresponduje s požadavky a které rýsuje obrysy datové základny budoucího IS. Mnohdy jsou entity reálného světa modelovány obecnějšími, více abstraktními, entitami. [11] Tento model je součástí této práce.
- Logický datový model (LDM) – je určitým mezistupněm mezi konceptuálním a fyzickým datovým modelem. Zůstává implementačně nezávislý. Precizuje objekty konceptuálního modelu, například entity mají uvedeny všechny atributy i identifikátory, relace „many-many“ jsou nahrazeny tabulkou. Pokud je u menších projektů obdobně precizován již konceptuální model (což je možné, pokud je u něj dosaženo dostatečné „laické“ srozumitelnosti a přehlednosti i při vyšší míře detailů), pak se z praktických důvodů tento logický datový model nesestavuje.[10] Tak je tomu i v této práci.

- Fyzický datový model (PDM) – specifikuje databázové tabulky a jejich sloupce, vztahy tabulek (reference) a obvykle i další objekty databáze. Plně odpovídá struktuře databáze, je tedy již implementačně závislý. Mnohdy ho lze získat z databáze (Reverse Engineer) a naopak je možné z něj databázi vygenerovat, přičemž si lze vybrat z celé řady systémů řízení báze data (např. Oracle, MS, SAP, …) Usnadňuje technickou orientaci v datových strukturách. [10] Tento model je součástí této práce.

Poměrně blízké datovým modelům jsou objektové modely, které bývají prezentovány pomocí UML (Unified Modeling Language) standardu a kde entitám (tabulkám) datového modelu odpovídají třídy objektů se vztahy vyjádřenými asociacemi.

3.2.2 Procesní modelování

Pojem proces je používán v poměrně veliké šíři významu, nicméně vždy se jedná o vývoj vlastností jednoho či více objektů včetně jejich vztahů v čase. [12] V souvislosti s řízením společností se objevuje pojem „Procesní řízení“. Alternativou k procesnímu řízení je dříve obvyklé tradiční funkční řízení založené na organizační pyramidě. V rámci procesního řízení se procesem rozumí zejména sled opakujících se činností s jasně definovaným vstupem a výstupem, dobou trvání a měřitelnými ukazateli. Součásti procesního řízení je:

- Popis procesů a stanovení cíle procesů.
- Dekompozice procesů.
- Procesní mapa.
- Definice vlastníků procesů.
- Měření a monitorování výsledků procesu.
- Optimalizace procesů.

V případě, že je popis procesů detailizován až do popisu elementárních úkonů včetně dotčených zdrojů, bývá někdy nazýván work-flow. Pojem work-flow se také používá jako název provádění jednoho konkrétního procesu. [8] Existují softwarové nástroje (například ARIS Toolset), které je možné nastavit dle konkrétního work-flow a které pak daný proces podporují. Podobně existují i komplexní informační systémy, které mají svou architekturu založenou na procesním pojetí.

Procesní řízení založené na popisu procesů zřejmě dalo základy i procesnímu modelování, které umožní sestavit model procesu vyjádřený grafickým diagramem, který usnadní porozumění procesu všem účastníkům. Pro oblast modelování procesů byla vyvinuta řada modelovacích metod a technik i řada notací pro jejich zobrazení, případně i jazyků pro jejich specifikaci, například:

- BPMN (Business Process Model and Notation) – soubor principů a pravidel, který slouží pro grafické znázorňování podnikových procesů pomocí procesních diagramů.

- DFD (Data Flow Diagram) – diagram datových toků, který lze užít pro modelování funkcí informačních systémů.
- Use Case Diagram – diagram případů užití, který zobrazuje chování systému tak, jak ho vidí uživatel. Obvykle využívá modelovací jazyk UML.
- Řada dalších: WS-BPEL (Web Services for Business Process Execution Language), …

Procesní modelování se stalo nedílnou součástí návrhu informačních systémů, kde přispívá k vyjasnění předpokládaných pracovních postupů při práci s budoucím IS mezi zadavatelem a vývojáři, ale také k bližšímu pochopení vztahů mezi jednotlivými funkcemi IS.

K tomuto účelu je v této práci zvolen v nástroji PowerDesigner model BMP (Business Process Model) typu (Model Type) „Business process diagram" s notací (Process Language) „Analysis", který je intuitivně srozumitelný všem účastníkům.

V rámci těchto BPM diagramů se proces modeluje jako posloupnost elementárních procesů (úkonů, operací), která je iniciovaná startovací událostí. Proces se ve svém průběhu může větvit v závislosti na specifikovaných podmínkách a naopak se může určitá větev synchronizovat s dalšími větvemi či s novou událostí. Každý proces, tedy i každá jeho větev, má své ukončení. Takto jsou koncipovány i diagramy, které jsou součástí této práce. [12]

V diagramech tohoto typu je možné dále uvádět organizační jednotky, a to i ve vztahu k jednotlivým elementárním procesům jako jejich iniciátory či respondenty. Toto se v této práci neuplatnilo, neboť se jedná o návrh takového IS, kde objekty typu organizační jednotky (skupiny uživatelů) budou teprve po instalaci systému uživatelsky specifikovány.

V diagramech tohoto typu je možné také uvádět úložiště dat, a to i ve vztahu k jednotlivým elementárním procesům se specifikací způsobu přístupu (create, read, update, delete), případně i s přesným odkazem na tabulky či jejich sloupce a s formátem komunikace. [12] Toto se v této práci neuplatnilo, neboť by došlo k nárůstu složitosti diagramů a snížení přehlednosti, což by negativně ovlivnilo jejich hlavní poslání – přispět během analýzy k vyjasnění předpokládaných pracovních postupů při práci s budoucím IS.

Na druhou stranu by to mohlo usnadnit komunikaci s programátory, takže nelze vyloučit, že před zahájením programátorských prací bude z daného BPM vygenerován (odvozen) další zpřesněný a detailnější, tedy již ne konceptuální, BPM, který bude provázán na PDM.

3.3 Použité nástroje

Pro návrh konceptuálního a procesního modelu budoucího IS „Hierarchický multilingvální diář" byl v rámci této práce použit CASE nástroj PowerDesigner 15.3., který byl použit i pro sestavení

fyzického datového modelu. Fyzický datový model byl sestaven pro systém řízení báze dat (DBMS – database management system, dále jen zkráceně „databáze“) SAP Sybase SQL Anywhere 12.

3.3.1 Nástroj CASE - PowerDesigner

PowerDesigner představuje komplexní modelovací nástroj softwarového inženýrství (CASE) zaměřený na návrh informačních systémů i na jejich rozvoj. Byl vyvinutý firmou Sybase, která je nyní vlastněna firmou SAP. [13] Pro účely této práce jsem měl možnost využít licenci vlastněnou firmou BM Servis. Volně k dispozici je PowerDesigner Free Viewer.

Jedná se o nástroj z oblasti MDE (Model-driven engineering), což je metoda vývoje software založená na tvorbě a využívání modelů. Podstatou návrhu a vývoje v PowerDesigner jsou tedy modely, je zde k dispozici několik typů modelů a lze je provazovat. Jedná se zejména o tyto modely:

- Model požadavků, přičemž jednotlivé strukturované požadavky lze provazovat na objekty všech dalších uvedených modelů ve smyslu „tento objekt přispívá k řešení daného požadavku“.
- Konceptuální datový model, ze kterého lze generovat (automaticky odvozovat) logický datový model, fyzický datový model a objektový model.
- Logický datový model.
- Fyzický datový model, ze kterého lze generovat skript pro vygenerování databáze.
- Business process model.
- Object-Oriented Model.
- Enterprise architecture.

Většina modelů se prezentuje formou grafických diagramů (jeden model může být prezentován několika diagramy), ve kterých jsou názorně a přehledně uspořádány symboly reprezentující klíčové objekty modelu. [13] V každém diagramu lze nastavit (Display Preferences) pro každý typ objektu, jak má vypadat jeho zobrazovaná podoba po stránce obsahu (Content) i po stránce formátování (Format). Uvedené nastavení způsobu zobrazení lze také aplikovat individuálně na každý objekt samostatně.

Souvislost mezi jednotlivými modely, která udržuje jejich konzistenci, je podporována pomocí:

- Generování modelu (Target Model) z jiného modelu (Source Model) při udržení vazeb mezi jednotlivými objekty, aby uživatel mohl rozhodnout o synchronizaci případných změn v obou modelech do cílového modelu. Například mezi fyzickým a konceptuálním datovým modelem.
- Specifikací „ShortCuts”, které mohou odkazovat na objekt jiného modelu. Například objekt „Data“ v BPM může být definován odkazem na objekt „Entity“ v CMD nebo na objekt „Table“ v PDM.
- Specifikací „Extended Dependencies“, které umožní vytvořit uživatelskou vazbu mezi libovolnými objekty libovolných modelů.

- Specifická vazba (Requirements) umožňuje specifikovat souvislost mezi jakýmkoliv objektem každého modelu a požadavky (z RQM – Requirements Model), k jejichž řešení přispěl.

V jednotlivých modelech i v jejich vazbách jsou k dispozici značné možnosti jejich přizpůsobení, a to nejen v ovládání, ale i z pohledu funkcionality, například dle interních metodik a standardů každého vývojového týmu.

PowerDesigner může pro svou práci používat jednotlivec i velké pracovní týmy, pro které se výhodně uplatní tyto vlastnosti:

- Veškerá data modelů sestavených v PowerDesigner mohou být ukládána a sdílena v Repository, tj. v definované SQL databázi, kde lze slučovat práci jednotlivců vývojového týmu.
- V Repository mohou být ukládány různé verze modelů či jejich objektů a ty mezi sebou netriviálně slučovány, což se uplatní zejména v situaci, kdy na jednom modelu provádí změny více osob.
- Pro přístup k modelům a jejich objektům z Repository lze definovat přístupová práva.
- Možnost plovoucí licence k PowerDesigner za použití licenčního serveru. Licenci lze přesunout na svůj notebook (Obtain Mobil Licence) a pak ji užívat bez připojení k licenčnímu serveru.

Z dalších zajímavých vlastností PowerDesigner ještě zmíním, že:

- Modely lze uspořádat v rámci WorkSpace do jednotlivých složek a projektů. Každý uživatel může míst několik svých různých WorkSpace (okno s přehledem modelů, složek a projektů).
- Jsou k dispozici výstupy představující dokumentaci modelu v několika verzích z hlediska obsahu, případně lze obsah uživatelsky specifikovat. Po stránce formátu může být výstup dokumentace v HTML anebo v RTF. Fakticky by tato dokumentace mohla nahradit kapitoly „Návrh řešení“ a „Realizace řešení“ této práce.
- Detailní, byť anglicky, dokumentace všech standardních funkcí PowerDesigner a zejména komplexní dokumentace jeho interního objektového modelu, která velmi usnadňuje realizaci netriviálních uživatelských rozšíření PowerDesigner.

Komplexnost a rozsah možností tohoto nástroje dokresluje i rozsah dokumentace – více než 3000 stran v PDF, dále metamodel jeho objektů v obdobném rozsahu, který je potřebný k programování uživatelských rozšíření. Vlastnosti jednotlivých modelů PowerDesigner, které jsou součástí této práce, jsou ještě detailněji zmíněny v relevantních kapitolách.

3.3.2 Databáze SQL Anywhere

SAP SQL Anywhere (původně Sybase SQL Anywhere) představuje relační systém řízení báze dat (DBMS – database management system), zkráceně databázi, který může být kromě platformy

Windows provozován i na dalších platformách. Její možnosti odpovídají požadavkům, jaké jsou na standardní SQL databáze kladeny, s řadou dalších rozšíření (webové služby, replikace, …). [14]

V rámci této práce byla použita tato databáze ve verzi „Sybase SQL Anywhere 12“. Byla v této verzi i specifikována ve fyzickém datovém modelu (Current Database), ze kterého byly automatizovaně založeny její objekty (například: Table, Column, Constraint, Index, Permission, View, Procedure, Event, User apod.).

Součástí tohoto DBMS je také Sybase Central (použit ve verzi 6.1), který umožňuje spravovat databázové objekty. Dále je součástí Interactive SQL (použit ve verzi 12.011), který umožňuje zadávat a provádět databázové příkazy včetně spouštění skriptů.

Tato databáze byla zvolena z toho důvodu, že ve firmě BM Servis s.r.o., která PowerDesigner pro tuto práci poskytla, vyvíjí informační systémy právě nad ní.

4 Analýza požadavků

Původní poměrně stručné požadavky z hlediska požadované funkcionality budoucího informačního systému byly analyzovány a na základě konzultací se zadavatelem a zejména na základě zpětné vazby z návrhu modelů byly zpřesněny do následující podoby.

Objektem se v této kapitole rozumí objekt reálného světa, o kterém budou v budoucím IS vedeny údaje (záznam v tabulce či tabulkách). Objekty jednoho typu se zde rozumí objekty reálného světa podobných vlastností, které budou v navrženém konceptuálním datovém modelu reprezentovány jednou entitou.

4.1 Diář a jeho obsah

Diářem se zde rozumí dle času uspořádaný seznam událostí diáře, které se vztahují k určitému zdroji (vlastníkovi diáře). V budoucím programovém řešení, zejména v závislosti na použitých nástrojích a prostředí, se rozhodne o způsobu prezentace diáře (text, grafika, text integrovaný do grafiky, …).

Zdrojem diáře se zde rozumí:

- Prostředek, například sdílená zasedací místnost nebo automobil.
- Uživatel, tj. fyzická osoba pracující s diářem (informačním systémem „Multilingvální diář“).
- Skupina uživatelů (například projektový tým, oddělení IT, …) vyjadřující jejich hierarchické uspořádání. Taková událost diáře se vztahuje na všechny členy (uživatele) skupiny, pokud ji explicitně nevyřadili.

Událost diáře je určena časovým intervalem (vymezený počátečním datem a časem a koncovým datem a časem) anebo časovým bodem (specifikovaný jen jedním datem a časem), dále místem konání (poznámka, GPS nebo adresa) a stavem, který charakterizuje, zda se jedná o navrženou, plánovanou anebo již zrealizovanou událost, případně zamítnutou či typovou. Význam události diáře je určen:

- Kategorií (např. pracovní porada IT oddělení, narozeniny, programování, jednání se zákazníkem, schůzka projektového týmu, …). Kategorie současně předává události diáře řadu implicitních vlastností včetně oprávnění.
- Popisem – krátký text charakterizující událost, který může být uveden ve více jazycích.
- Komentář – detailní popis, který může být uveden ve více jazycích.

Jedna událost diáře se může týkat celé řady zdrojů. Některé události mohou mít díky svému významu nastaveny signalizace (zpráva emailem, barva zobrazení, upozorňovací událost) určeným uživatelům či jejich skupinám.

4.2 Hierarchie (sdílení diářů)

Hierarchií se zde rozumí především možnost zařazovat uživatele do skupin uživatelů (a také skupiny uživatelů do skupin uživatelů), pak je pohodlně umožněno pracovat s celou skupinou uživatelů najednou. V takovém případě přebírá uživatel z nastavení pro skupinu uživatelů (dědí na základě členství ve skupinách):

- přístupová oprávnění,
- přiřazení události diáře skupině (bude mít i ve svém diáři),
- signalizace události diáře skupině (bude i jemu signalizováno).

Úrovně přístupových oprávnění musí umět zajistit, že uživatel:

- Neví o existenci objektu, resp. o existenci objektů daného typu, nemá-li žádné oprávnění.
- Vidí jen existenci objektu, resp. existenci objektů daného typu (obvykle formou stručného seznamu), ale nevidí jeho detailní údaje (význam); například vidí, že nadřízený má určitý časový interval obsazen, ale nevidí čím.
- Vidí detailní údaje objektu, resp. objektů jednoho typu.
- Může použít daný objekt pro akce (například může událost diáře vztáhnout k danému zdroji, má-li tuto úroveň oprávnění k danému zdroji), resp. může zakládat objekty daného typu.
- Uživatel může objekt, resp. objekty daného typu, plně spravovat.

Úroveň oprávnění daného uživatele, člena skupin uživatelů, se vyhodnotí dle těchto priorit:

- Má přiřazeno explicitní oprávnění, tj. oprávnění je přiřazeno přímo uživateli.
- Nejvyšší úroveň oprávnění vyplývající z členství ve skupinách, do kterých je zařazen.
- V případě, že daná skupina nemá explicitně specifikované oprávnění, stanoví se ze skupin, do kterých je zařazena, a to rekurzí až k vrcholové skupině.
- Pokud není identifikováno explicitní oprávnění pro uživatele ani pro žádnou jeho skupinu, pak se jedná o nejnižší úroveň oprávnění, kdy neví o existenci objektu.

Ke všem objektům jsou oprávnění specifikována zprostředkovaně dle typu objektu. Oprávnění přímo pro jednotlivé objekty lze specifikovat ke zdrojům, kategoriím, alokacím a šablonám. K události diáře se úroveň oprávnění vyhodnotí jako nejnižší oprávnění z oprávnění k:

- Kategorii,
- Zdroj,
- Alokace.

Sdílením diářů se zde rozumí, že jediná událost diáře se bude prezentovat v diáři všech zdrojů, kterých se týká.

4.3 Multilingvální systém

Navrhovaný informační systém bude možné provozovat současně v několika jazycích, bude tedy multilingvální. Všechny vstupované texty musí být možné zadat, evidovat a publikovat v několika jazykových verzích.

Jazyky, ve kterých lze IS provozovat (tj. ve kterých lze evidovat popisné údaje), lze spravovat uživatelsky. Tedy je možné:

- Přidat další jazyky; předpokládá se instalace v českém jazykovém prostředí a další jazyky bude možné přidávat správcem IS; ve všech evidovaných jazycích je možné informace vkládat či prezentovat.
- Definovat implicitní jazyk IS, který bude implicitně přiřazován novým uživatelům.
- Přiřazovat uživateli jeho jazyk, ve kterém bude s diářem implicitně pracovat.
- Specifikovat pro jednotlivé typy informací, ve kterých jazycích musí být evidovány povinně jejich textové údaje. Systém nedovolí zapsat takovéto informace, pokud nebudou uvedeny ve všech povinných jazycích.
    - V případě typu informace „Událost diáře“ specifikovat povinný jazyk nastavením povinného jazyka pro každou kategorii samostatně.

Pro zobrazení každého textového údaje se volí jazyk v tomto pořadí:

- V jazyce nastaveném uživatelem.
- V implicitním jazyce uživatele.
- V povinném jazyce pro danou kategorii diáře s nejvyšší prioritou – uplatní se jen v případě typu objektu „Událost diáře“.
- V povinném jazyce pro daný typ objektu s nejvyšší prioritou.
- V nepovinném jazyce s nejvyšší prioritou.

4.4 Signalizace

4.4.1 Signalizace kolizí

Kolizí se zde rozumí situace (a to i potenciální), kdy se časový interval vymezující trvání jedné události diáře překrývá s jinými událostmi diáře v diáři některé dotčeného zdroje.

Pokud při zobrazení diáře existuje kolize, tak se zobrazí specifickým druhem písma nastaveným v parametrech uživatele (implicitně tučně). Pokud při zápisu nové události diáře (nebo při přiřazení nových zdrojů k dané události diáře) hrozí některým zdrojům kolize, řídí se chování systému dle nastavení pro danou kategorii události diáře (při kolizi dvou různých kategorii je rozhodující ta s nižším nastavením). Možná nastavení:

- Událost nebude zapsána, a tedy nebude přiřazena ani k jedinému zdroji, tedy se vůbec nic nezapíše. Uživatel bude upozorněn.
- Událost bude zapsána, ale bude přiřazena jen ke zdrojům, u kterých nenastane kolize. Uživatel bude upozorněn.
- Událost bude zapsána, přiřazena ke všem zdrojům a uživatel bude upozorněn na kolizi.
- Událost bude zapsána, přiřazena ke všem zdrojům a uživatel nebude upozorněn na kolizi.

4.4.2 Signalizace významných událostí

Bude možné na některé (důležité, významné) události diáře anebo na změny v nich upozornit relevantní uživatele některou z těchto forem signalizace:

- Barevným zobrazením dané události v diáři uživatele.
- Odesláním upozorňovacího mailu.
- Prezentovat v diáři upozorňovací časový bod vztažený ke zdrojové události.

Signalizaci bude možné přednastavit automaticky podle dané kategorie události diáře s tím, že je možné ji pro konkrétní událost diáře pozměnit.

Pro signalizaci události bude možné specifikovat, jakou událostí (založení záznamu, oprava, …) se aktivuje, k jakému času se vztahuje (čas zahájení nebo ukončení události, čas zápisu, čas opravy apod.) a jaký bude posun od něj. Signalizace a její způsob může záviset na stavu události (navržená, plánovaná, …).

4.5 Použití šablon

Šablona představuje nástroj, pomocí kterého lze pohodlně najednou vygenerovat několik událostí diáře a současně je přiřadit k řadě zdrojů s nastavením signalizace. Její použití lze očekávat například pro:

- Vygenerování každoročních oslav narozenin ze zvolené události „narozeniny“ konkrétního pracovníka na příštích *n* roků s udržením vazby na původní událost. Vazba následně umožní další funkce, např. zrušení všech navazujících záznamů od aktuálního data apod.
- Vygenerování návrhu porady projektového týmu na zadaný termín s automatickou signalizací všem relevantním zdrojům, například na základě typové události diáře, která nese všechny relevantní vazby.

4.6 Data

Vzhledem k tomu, že se požadují významné uživatelské možnosti konfigurace budoucího IS a jeho funkcí, je potřebné, aby řada z nich byla řízena daty. Vzhledem k předpokladu aplikace v různém primárním jazykovém prostředí se požaduje, aby tato data byla základním způsobem nastavena při instalaci systému.

Zejména je nutné při instalaci nastavit (automaticky založit data):

- Typy objektů a všechny další programátory nadefinované číselníky.
- Uživatele „Správce“, který má oprávnění administrátorské ke všem typům objektu.
- Skupinu uživatelů „Každý uživatel systému“, do které bude každý uživatel automaticky zařazen.
- Primární jazyk a jeho přiřazení správci a nastavení jeho povinnosti pro všechny typy objektu.
- Názvy číselníků v primárním jazyce. Pokud primární jazyk nepatří mezi programátory předdefinované, musí se při instalaci zadat (překlad z anglického jazyka).
- Parametry systému.

Dále je vhodné při instalaci nastavit (automaticky založit) data představující vzorová uživatelská nastavení, např.:

- Barvy písma a výplně pro zobrazení událostí diáře.
- Způsoby signalizace.
- Kategorie událostí diáře, včetně způsobu řešení kolizí, signalizace
- Šablony.
- Parametry systému.

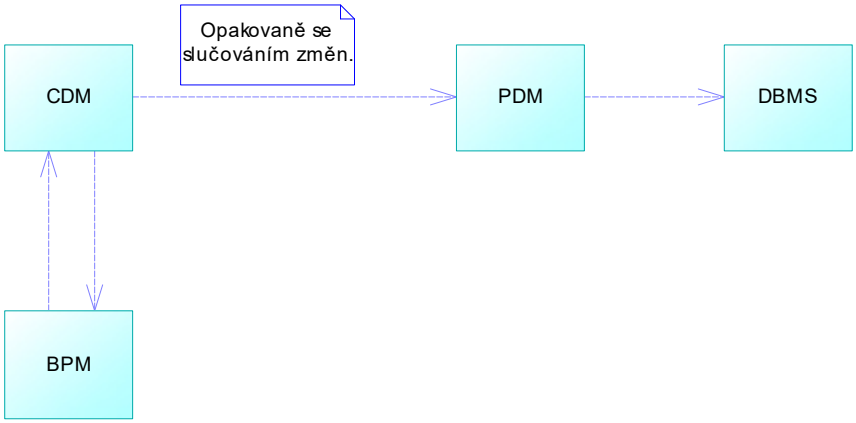
5 Návrh řešení

Návrh informačního systému „Multilingvální diář“ pomocí modelovacího nástroje PowerDesigner (PD) je reprezentován dvěma modely:

- Konceptuální datový model (CDM).
- Business proces diagram (BPM).

CDM a BPM jsou modely představující návrh systému v konceptuální úrovni. Testováním jejich vzájemné integrity a testováním oproti požadavkům lze prověřit ucelenost a kvalitu návrhu při současném zpřesňování požadavků. Toto se finálně provádí ve spolupráci se zadavatelem požadavků (autorem požadavků).

Z dokončeného a odsouhlaseného CDM lze v PD pokročit od návrhu k realizaci tím, že se automatizovaně vygeneruje fyzický datový model (PDM) a následně i databáze (resp. databázové objekty), jak je patrné z obrázku.



Obrázek 1- Návrh postupu

PD současně disponuje poměrně sofistikovanými nástroji pro synchronizaci CDM a PDM během celého životního cyklu navrženého a realizovaného informačního systému. Tato synchronizace je založena na vazbě mezi zdrojovými objekty CDM a cílovými objekty PDM, takže případné změny objektů ve vygenerovaném cílovém PDM (například názvy tabulek, atributů, doplněné indexy apod.) jsou nadále provázané na původní zdrojové objekty CDM. Synchronizace tedy umožní při opakovaném generování PDM zachovat v něm užitečné změny a současně ho doplnit o užitečné změny ve zdrojovém CDM.

Tato kapitola „Návrh řešení“ má zejména doprovodný charakter, neboť vlastní návrh řešení představují přiložené modely CDM a BPM.

5.1 Použitá metodika

Pro návrh IS „Multilingvální diář“ byla použita metodika, kdy se na základě uživatelských požadavků (zadání) navrhuje konceptuální datový model, který se systematicky testuje pomocí procesního modelu, což vede na změny v jeho návrhu. Tyto modely, jako zadavateli přiměřeně srozumitelný návrh řešení, jsou s ním konzultovány, což vytváří zpětnou vazbu na požadavky (resp. analýzu požadavků), takže se zpřesňují.

Po úplném vyladění CDM a BPM se přejde vlastnímu vývoji informačního systému programátory (developery), kdy prvním krokem je sestavení fyzického datového modelu (tj. jeho automatické odvození z CDM) s následným automatickým vygenerováním databáze.

Pro tvorbu jednotlivých modelů byla převzata metodika z dokumentace PowerDesigner, přičemž zde je uváděn její stručný výtah.

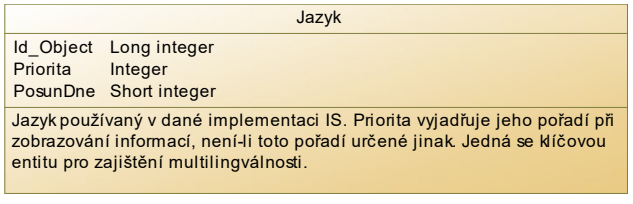
5.2 Konceptuální datový model (CDM)

V tomto případě jsou zde analyzovány a navrženy ty entity, které souvisí s komplexním vedením diářů ve více jazycích včetně jejich sdílení pracovními týmy. Entity jsou v tomto modelu uspořádány v diagramech dle standardů PD. Návrh CDM je zaměřen na logiku budoucí aplikace, tj. na informační schopnost, která je odpovědí na požadavky, nikoliv na formu prezentace dat.

5.2.1 Objekty konceptuálního datového modelu v PowerDesigner

Jádrem tohoto modelu jsou diagramy, ve kterých jsou entity propojeny relacemi (vyjadřují vztahy entit). Pro jejich zobrazení je v tomto návrhu použita notace „Entity/Relationship“. V diagramech prezentujících model se používají tyto grafické značky (Symbols) pro jednotlivé druhy objektů:

- Entity (entita) – reprezentuje typ objektu, o kterém mají být uchovávány informace (například „Jazyk“, „Adresa“, „Skupina uživatelů“, „Uživatel“, …). V diagramech CDM této práce je pro entity nastaveno (Display Preferences), aby se zobrazoval tento obsah: Název, všechny atributy včetně datového typu a komentář. Z entit konceptuálního datového modelu se generují tabulky do fyzického datového modelu. [11]



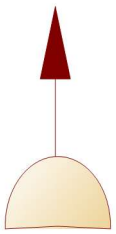
Obrázek 2- Entita

- Relationships (relace) – reprezentuje vztah mezi dvěma entitami s určitým významem, například mezi entitami „Adresa“ a „Stát“ je relace „Adresa je v daném státu“. Součástí relace je specifikace kardinality, tj. kolik vztahů může vytvářet každý výskyt entity s druhým maximálně

Obrázek 3- Relace

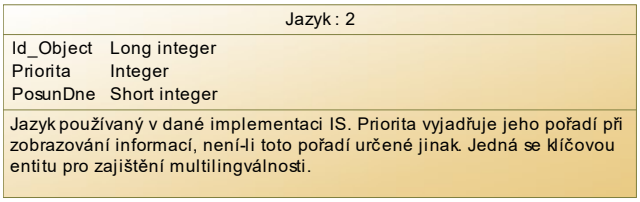
a kolik vztahů musí vytvářet minimálně. Například stát může být uveden v libovolném počtu adres anebo v žádné adrese, a naopak adresa musí patřit právě do jednoho státu. Kardinalita je graficky znázorněna na obou stranách relace. Pokud je minimální i maximální kardinalita 1, pak se zobrazí jen jednou „čárou“. [11]

- Inheritance (dědičnost) – reprezentuje relaci, která definuje jednu entitu „dítě“ (Child) jako zvláštní případ obecnější entity „rodič“ (Parent). (8) Například entita „Uživatel“ (Child) je zvláštní případ entity „Zdroj diáře“ (Parent) podobně jako „Prostředek“ (Child) a „Skupina uživatelů“ (Child). Součástí této relace je informace (Complete), zda každý výskyt v rodičovské entitě (Parent) patří do některé z uvedených entit (Children), a informace (Mutually exclusive children), zda každý „rodič“ (Parent) má své „dítě“ právě jen v jediné „dětské“ entitě (Child).



Obrázek 4 - Dědičnost

Pokud je z důvodu přehlednosti vhodné v určitém diagramu zopakovat určitý objekt (obvykle entitu), lze použít techniku „Paste as Shortcuts“. Pak se jedná stále jen o jeden objekt modelu, ale ten je v diagramech prezentován dvěma či více symboly. V nadpisu symbolu se objevuje jeho pořadí, například „Jazyk: 2“.



Obrázek 5 - Entita jako Paste as Shortcuts

Jak již bylo zmíněno dříve, lze modifikovat způsob zobrazení jednotlivých objektů v diagramu. V této práci je uplatněno několik změn implicitního nastavení (Display Preferences) pro zobrazení objektů CDM v diagramech, například:

- V některých diagramech, kde to přispěje k celkovému pochopení návrhu, se zobrazují názvy relací (Name of Relationship) a v jiných nikoliv.
- Nastaveno zobrazování komentáře všech entit (Comment of Entity), aby se vypovídací schopnost diagramů zvýšila.
- Nastavena jiná barva výplně i ohraničení většiny objektů.

Dále jsou v této práci specifikovány další objekty CDM, které se v diagramech zobrazují jen částečně v rámci symbolu entity anebo se v nich nezobrazují vůbec:

- Data Item – nezávislá elementární informace, např. čas „Platné v období od“. Není generována do fyzického modelu, ale usnadní modelování a jednotné pojmenování obdobných atributů (následně sloupců tabulek). [11]

- Attribute – vlastnost entity, např. Data Item „Platné v období od“ je aplikované s významem, od kdy platí zařazení uživatele do skupiny, když je relevantní Data Item připojeno k entitě „Příslušnost do skupiny“. Určitý počet atributů se může zobrazovat v rámci symbolu entity. [11]
- Domain – doména umožnuje specifikovat datový typ, který lze opakovaně používat jako typ pro Data Item. [11] To umožní, že se datový typ dané položky může v pokročilejší fázi návrhu pohodlně změnit (případně může být změněn až v návazném vygenerovaném PDM).
- Identifier – identifikátor, jako vlastnost entity, je specifikován atributy entity. Hodnota identifikátoru (jeho atributů) unikátně identifikuje každý objekt dané entity. Jeden z identifikátorů je označen za primární. Určitý počet atributů se může zobrazovat v rámci symbolu entity. [11]

Mezi klíčové vlastnosti všech objektů (resp. valné většiny) patří:

- Name – jméno objektu, pod kterým je prezentován i v diagramu. To by mělo prezentovat klíčový význam objektu a být srozumitelné i pro netechnické pracovníky (např. zadavatel).
- Code – technické jméno objektu, které je následně použito v PDM při generování databázových objektů jako jejich jméno. Obvykle silně zkrácené a bez mezer.
- Comment – komentář objektu specifikuje význam objektu volným textem.
- Generate – zda má daný objekt vstoupit do generování cílového modelu. To umožní v analytickém CDM použít objekty, které přispějí k jeho větší vypovídací schopnosti, ale nemají technický význam.

5.2.2 Úvodem k CDM multilingválního diáře

Klíčové požadavky na navrhovaný IS „Multilingvální diář“ (multilingvální, hierarchický, schopný integrace) si vynucují, aby celé řešení, a tedy i data (entity, tabulky), mělo jednotný rámec, ve kterém je bude možné „pohodlně“ vyřešit pro jakákoliv data budoucího IS.

Jako jádro celého řešení „multilingválního diáře“ jsem zvolil takovou koncepci struktury entit, kdy všechny entity mají společného jediného rodiče (Parent Entity), a to entitu „Object“. V takovém případě platí, že atributy a relace rodičovské entity jsou současně zděděny jejich „syny“. Podobně tomu bude i ve fyzickém datovém modelu a následně i jakýkoliv záznam v jakékoliv tabulce s daty budoucího IS bude mít svého „otce“ (záznam tabulce odpovídající entitě „Object“), který ponese údaje (sloupce), které jsou společné všem záznamům. Všechny entity tedy mají jednotný primární identifikátor „Id_Object“, který dědí z entity „Object“.

Toto řešení zajistí jednotným způsobem specifikovat popis všech objektů (jméno, zkratka, další text a komentáře) v různých jazycích. Podobně lze jednotně specifikovat oprávnění i další případnou funkcionalitu.

Uvedená koncepce je patrná z diagramu „Koncept objektů“, kde jsou dále všechny entity s cílem usnadnit orientaci v nich rozděleny jednotlivými inheritancemi do těchto skupin:

- Silné klíčové entity - ty jsou jádrem celého tohoto návrhu. Reprezentují obvykle významné objekty reálného světa, o kterých je v IS vedena evidence. Vstupují mezi sebou do vztahů (přímo anebo pomocí vazebních entit). K popisu jejich vlastností se mimo jiné používají číselníky. Každá z těchto entit vstupuje do řady vazeb a je jí v tomto návrhu obvykle věnován samostatný diagram.
- Číselníky definované programátory – ty představují pevnou množinu vlastností (i jejich hodnot) přidělovaných jiným entitám, následně tedy objektům evidovaným v IS. Toto přidělení vlastnosti již může provádět uživatel. S těmito číselníky (resp. jejich hodnotami) je obvykle spojen programový kód. Specifikace těchto číselníků formou entit a následně formou hodnot sloupců v tabulkách budoucího IS zvyšuje celkovou přehlednost a usnadní budoucí rozšiřování IS.
- Číselníky uživatelsky definovatelné – ty představují takové vlastnosti, jejichž hodnoty může uživatel specifikovat a přidělovaných jiným entitám, následně objektům v IS. Například barva písma apod.
- Vazební entity – ty vyjadřují vztahy mezi entitami a přiřazují danému vztahu určité vlastnosti. Nejčastěji se jedná o vztahy mezi silnými entitami.

Určitou výjimku z uvedené základní koncepce představuje z důvodu referenční integrity entita „Typ objektu“. Dále jsou výjimkou entity „Text v jazyce“ a „Komentář v jazyce“, které mají jen popisný charakter s cílem dosáhnout multilingválního řešení a které tedy nejsou objektem diáře v pravém slova smyslu (oprávnění apod. vyplývá z objektu, ke kterému se vztahují).

5.2.3 Konstrukce CDM s přihlédnutím ke generování PDM

Při konstrukci CDM se předpokládalo, že z CDM bude generován PDM a následně i databáze (skript pro vytvoření databáze). PD udržuje vazbu mezi objekty zdrojového CDM a generovaného cílového PDM, takže je možné při každém generování PDM rozhodnout, jakým způsobem se má spojit (merge) model CDM a PDM, třebaže v nich byly provedeny změny. Viz též kapitola „Generování PDM z CDM“.

Spolupráce analytika a programátora tedy nekončí jednosměrným předáním CDM programátorům, ale pokračuje po celý životní cyklus navrženého a realizovaného informačního systému.

Při generování PDM z CDM (Generate Physical Data Model) uskutečňuje PD transformace těchto typů objektů CDM na uvedené typy objektů PDM:

CDM	**PDM**
Diagram	Diagram
Entita	Tabulka
Atribut entity	Sloupec tabulky
Doména	Doména
Identifikátor	Klíč
Relace	Reference + sloupce cizího klíče + index cizího klíče
Relace (rodič-dítě)	Reference + sloupce primárního klíče + primární index

Bylo tedy nutné/vhodné již při tvorbě CDM uplatnit některé zásady, aby synchronizace vztahu mezi CDM a PDM nevyžadovala příliš mnoho zásahů. V tomto návrhu CDM jsou tedy uplatněny tyto zásady:

- CDM je konstruováno v poměrně velkém detailu (vazební entity, číselníky, …), takže ve vygenerovaném PDM se bude aplikovat minimum úprav. Fakticky entity CDM budou plně odpovídat tabulkám PDM, atributy CDM budou plně odpovídat sloupcům PDM.
- Každý objekt CDM je pojmenován lidsky srozumitelným českým názvem (Name), kterým se prezentuje i v diagramech, tedy přispívá k porozumění návrhu zadavatelem. Technické jméno (Code) všech objektů PD je uváděno rovnou na úrovni CDM stručným anglickým názvem (např. DiaryEvent). To se prezentuje v diagramech PDM a je z něj následně odvozen název databázového objektu.
- Vzhledem k tomu, že otcem (Parent Entity) všech entit je jediná entita „Object“, není nutné v ostatních entitách uvádět primární identifikátor ani jeho atributy, neboť budou zděděny z entity „Object“ a při generování PDM budou automaticky založeny odpovídající sloupce tabulek i primární klíče.
- Podobně budou při generování PDM automaticky z relací v CDM odvozeny a založeny sloupce odpovídající cizím klíčům včetně indexů. Není tedy v CDM nutné uvádět atributy představující cizí klíč ani odpovídající identifikátory.

Třebaže je CDM konstruováno s přihlédnutím k možnostem generování PDM, i tak je nutné určitá nastavení definovat až v PDM (viz kapitola „Nastavení PDM“). Na druhou stranu je zřejmé, že

v situaci, kdy CDM navrhuje čistokrevný analytik, tak se vůbec nezajímá o to, jak s modelem dále pracují programátoři.

5.2.4 Objekty CDM bez symbolu

Součástí návrhu CDM jsou také typy objektů, které se v diagramu nemohou prezentovat vlastním symbolem, někdy je možné je zobrazit, alespoň částečně, v rámci symbolu entity. I tyto objekty ovlivňují generování PDM a následné generování skriptu pro vytvoření databáze. Jsou zde použity tyto typy objektů CDM: Domény, datové položky, atributy a identifikátory.

5.2.4.1 Domény

V rámci tohoto návrhu se používají tyto domény:

- ID – určená pro jednoznačné identifikátory všech entit, které mají jako otce (Parent Entity) entitu „Object“. Nyní je nastavena na datový typ „Long integer“.
- Ano/Ne – určená pro jednoduchou logickou hodnotu TRUE nebo FALSE. Nyní je nastavena na datový typ „Short integer“ s možností hodnoty Ano=1 nebo Ne=2.
- Komentář – určená pro libovolně dlouhý text. Použito pro komentáře a specifikace výrazů. Nyní je nastavena na datový typ „Text“.
- Název – určená pro text střední délky. Použito pro názvy, řádky adresy, hodnoty parametrů, … Nyní je nastavena na datový typ „Characters (256)“.
- Programátorská zkratka – určená pro symbolická jména, které se uplatní v programovém kódu nebo v uživatelské specifikaci výrazů. Použito například pro symbolická jména parametrů šablony, identifikátor typů objektů apod. Nyní je nastavena na datový typ „Characters (32)“.
- Čas dne – určená pro určení času v rámci dne, tedy času bez data. Použito například při specifikaci obvyklé disponibility zdroje v rámci daného dne týdne. Nyní je nastavena na datový typ „Time“.
- Časový bod – určená pro vyjádření data a času. Nyní je nastavena na datový typ „Timestamp“.

Hlavním výhodou použití domén je dosažení určité standardizace datových položek (Data Items a následně Columns) a zejména jednoduchá (na jediném místě) změna datového typu či jeho rozsahu, tedy i dosažení větší nezávislosti CDM na datových typech databáze, kterými bude disponovat použitá databáze. Například: Pokud cílová databáze umožní definovat sloupce s datovým typem „Logical“, pak bude vhodné u domény „Ano/Ne“ zaměnit datový typ „Short integer“ na „Logical“.

5.2.4.2 Data Item, Attribute

V rámci tohoto návrhu jsem použil jednu elementární datovou položku (Data Item) pro specifikaci více atributů různých tabulek, tedy celkový význam atributu je pak dán významem datové položky a významem entity.

Například datové položky „Platné v období od“ a „Platné v období do“ je použita jako atribut v entitách „Obvyklá disponibilita zdroje“ a „Příslušnost do skupiny“. V obou případech specifikují období platnosti, jednou pro disponibilitu zdroje a jednou pro členství uživatele ve skupině uživatelů.

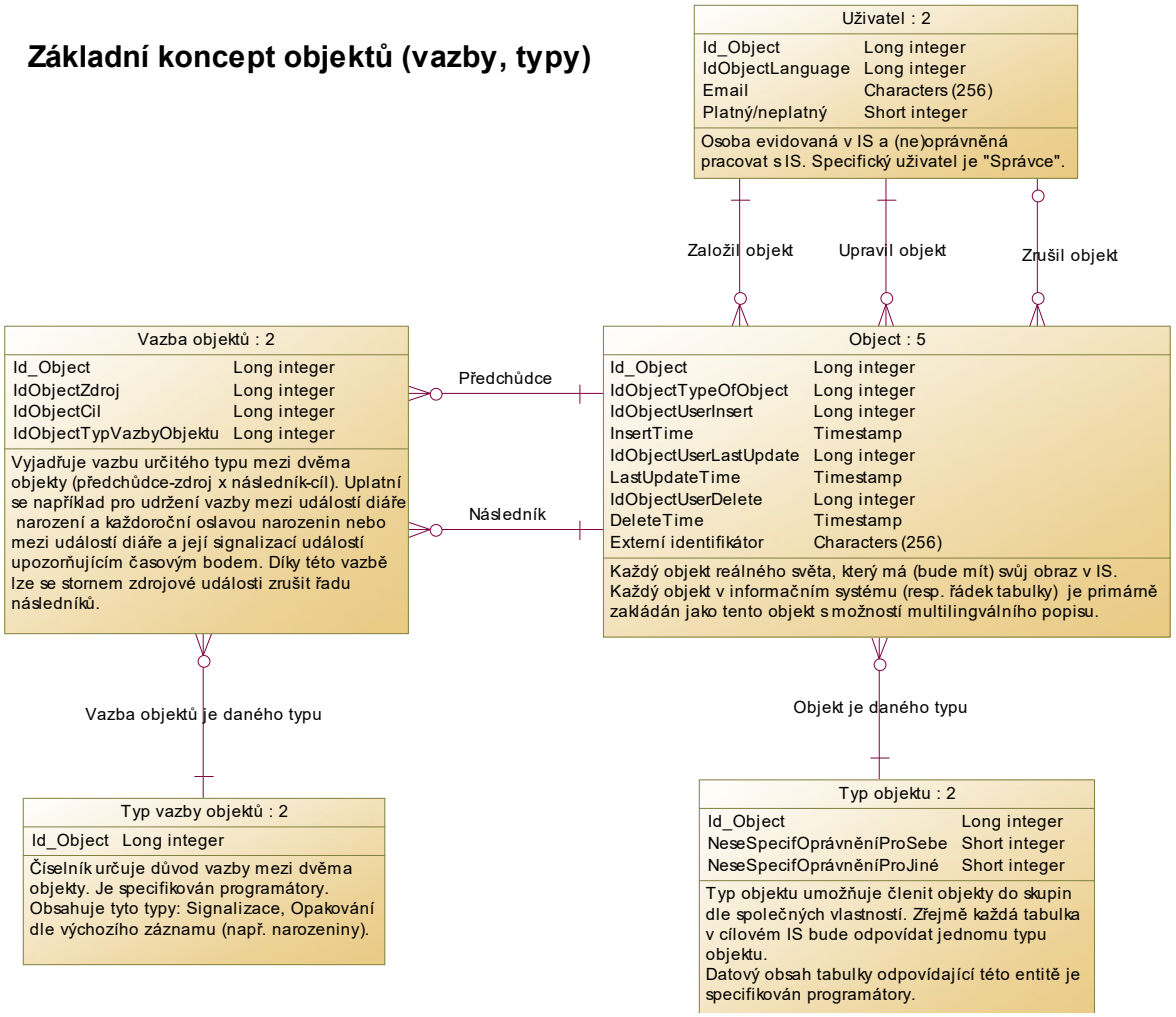
5.2.4.3 Identifikátory

V rámci tohoto návrhu je použit identifikátor „IdObj“ (doména ID), který jednoznačně identifikuje výskyty v entitě „Object“. Tedy i ve všech entitách, pro které je entita „Object“ otcem (Parent Entity). Výjimku tvoří tyto entity:

- „Typ objektu“, kde je identifikátorem její atribut „IdTypeOfObject“.
- „Text v jazyce“, kde je identifikátorem její atribut „Pořadí textu“. Ten bude ovšem v PDM ještě doplněn o atributy, které vyplývají z relací jako cizí klíče a které identifikují popisovaný objekt a použitý jazyk.
- „Komentář v jazyce“, kde je identifikátorem její atribut „Pořadí komentáře“. Ten bude ovšem v PDM ještě doplněn o atributy, které vyplývají z relací jako cizí klíče a které identifikují komentovaný objekt a použitý jazyk.

5.2.5 Diagram CDM – Koncept objektů

V tomto diagramu je prezentována základní koncepce jednotných objektů, kdy všechny entity mají svého rodiče (Parent entity), entitu „Object“. Každý záznam v ní bude jednoznačně identifikovatelný pomocí identifikátoru „Id_Object“, kterým se bude realizovat vazba se svým synem (Children entity).



Obrázek 6- Základní koncept objektů

Každý „Object“ je určitého typu, což je v diagramu vyjádřeno relací k entitě „Typ objektu“ v první části tohoto diagramu. Obsah entity „Typ objektu“ není uživatelsky definovatelný a je s ním spojena relevantní programová obsluha (programový kód, tabulka v databázi, způsob specifikace přístupových práv apod.). Data v tabulkách budoucího IS odpovídající této entitě musí být automaticky vytvořena při instalaci systému. Lze předpokládat, že každé entitě (následně tabulce) bude odpovídat jeden typ objektu.

„Typ objektu“ je uživateli přístupný pro definici přístupových práv, specifikaci povinnosti jazyka, pro specifikaci jeho názvu v jazyce, komentáře v jazyce apod. Dále každý typ objektu nese informaci, zda lze definovat oprávnění k jednotlivým objektům daného typu a zda se toto oprávnění uplatní:

- Přímo pro přístup k danému objektu (například pro typ objektu „Šablona“ bude nastaveno na ANO, což umožní definovat přístup ke každé šabloně individuálně).
- Zprostředkovaně k objektům, které jsou k danému objektu ve významné vazbě (například pro typ objektu „Alokace“ bude nastaveno na ANO, což umožní zprostředkovat přístup k událostem diáře, ve kterých je uvedena daná alokace).

Tato rodičovská entita „Object“ poskytuje všem entitám tyto jednotné atributy:

- Jednoznačný identifikátor.
- Typ objektu.
- Uživatel, který daný objekt založil.
- Časové razítko, kdy byl objekt založen.
- Uživatel, který daný objekt naposledy opravoval.
- Časové razítko, kdy byl objekt naposledy opraven.
- Uživatel, který daný objekt zrušil.
- Časové razítko, kdy byl objekt zrušen.

Předpokládá se, že zrušení objektu bude realizováno záznamem času jeho zrušení a uživatele, který ho zrušil (nikoliv fyzickým zrušením záznamu). Následně již s tímto objektem není možné standardně pracovat.

Každý objekt také vstupuje do vztahů s jazykem (entita „Jazyk“), viz diagram „Multilingvální řešení“, standardním způsobem.

Kromě specifických vazeb navržených formou relací či vazebních entit může v případě potřeby každý objekt vstupovat do vazby (vztahu, relace) s jakýmkoliv jiným objektem, a to i opakovaně. Tato vazba je realizována entitou „Vazba objektů“. Součástí této vazby je specifikace typu vazby (entita „Typ vazby objektů“). Data odpovídající entitě „Typ vazby objektů“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému. V tomto rozsahu návrhu předpokládáme dva typy vazby:

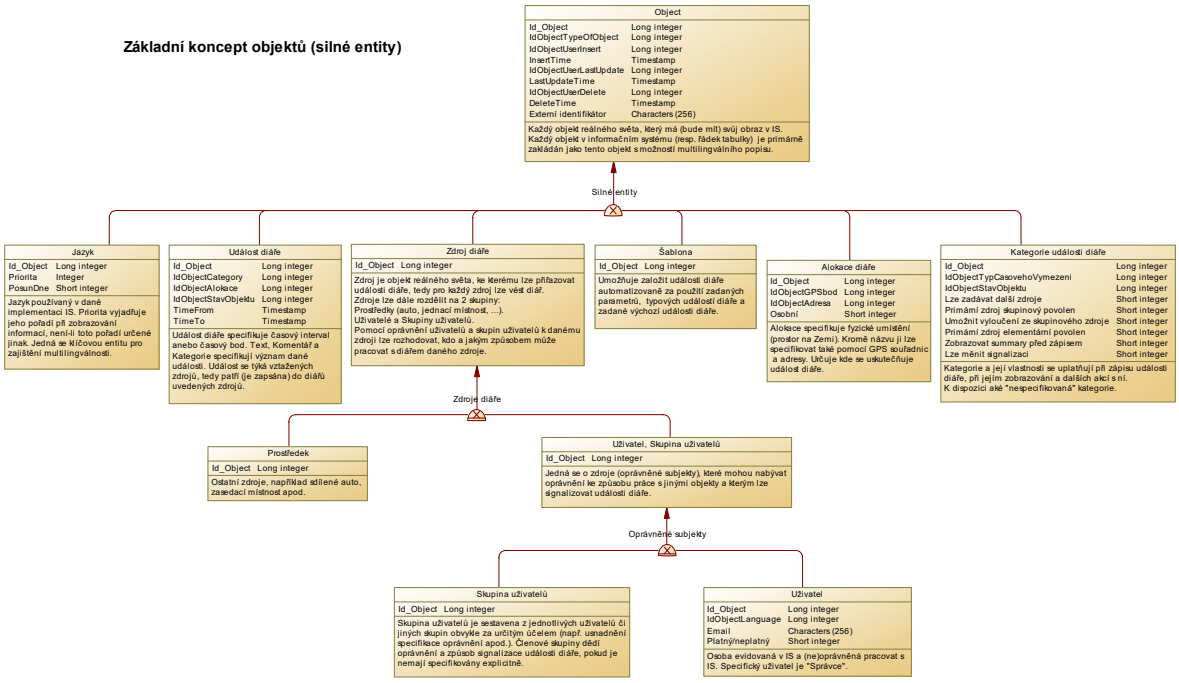
- Signalizace – zdrojová událost diáře je signalizována cílovými událostmi diáře (upozorňujícími časovými body).
- Opakování – cílové události diáře opakují zdrojovou událost diáře (např. narozeniny).

Lze očekávat, že s rozšiřování funkcionality navrhovaného IS se objeví nové typy vazeb mezi objekty. Takto navržené řešení vazeb objektů toto rozšíření usnadní.

Dále tento diagram specifikuje a prezentuje příslušnost k rodičovské entitě „Object“ (Parent) platnou pro všechny navržené entity (Children). Explicitní specifikace této příslušnosti v CDM zajistí odpovídající vygenerování PDM s referencemi. Tato příslušnost je zde specifikována pomocí těchto Inheritancí (každá představuje samostatnou část tohoto diagramu):

- Silné (klíčové) entity.
- Číselníky definované programátory.
- Číselníky uživatelsky definovatelné.
- Vazební entity.

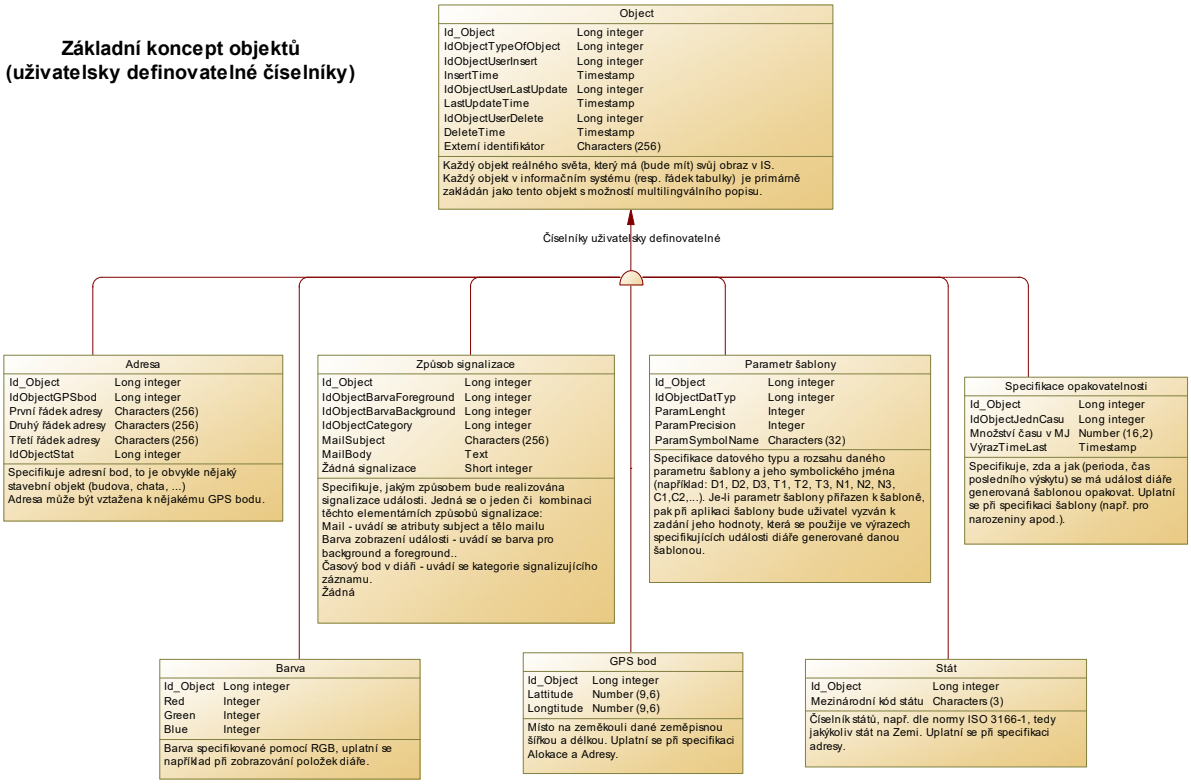
Každá z těchto Inheritancí odpovídá jedné skupině entit a slouží pouze pro přehlednější zobrazení (pro správné vygenerování PDM by stačila jediná Inheritance). Opakované zobrazení entity „Object“ je zde realizováno pomocí Shortcuts, tedy se jedná o jediný objekt PD prezentovaný opakovaně více symboly.



Obrázek 7- Základní koncepce objektů (silné entity)

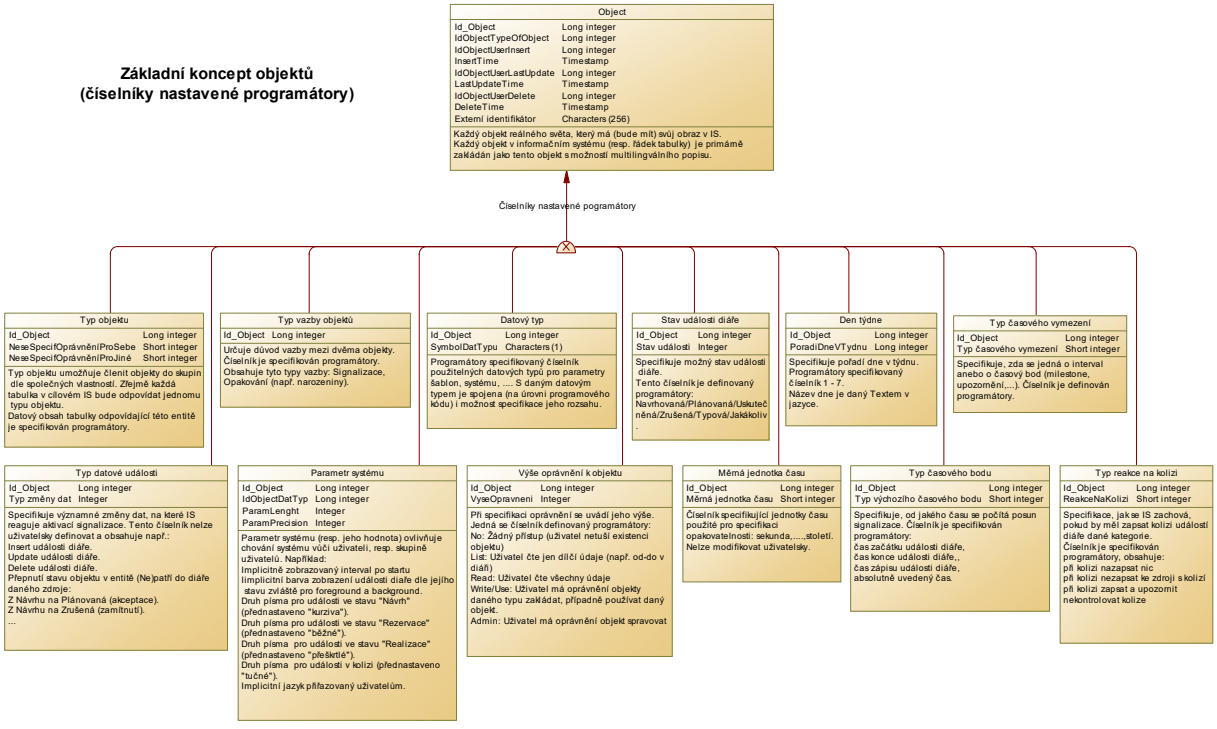
V části diagramu s Inheritancí „Silné entity“ se prezentuje příslušnost k rodičovské entitě „Object“ (Parent) všech navržených silných entit (Children). Je zde patrné, že entita „Zdroj diáře“ má dále své syny (Children). Toto je detailně popsáno v diagramu „Zdroje diáře“, zde jen uveďme, že i v tomto případě budou všechny entity mít jako svého (pra)otce entitu „Object“.

V části diagramu s Inheritancí „Číselníky uživatelsky definovatelné“ se prezentuje příslušnost k rodičovské entitě „Object“ (Parent) všech navržených entit (Children) tohoto typu. Některé z těchto číselníků mají elementární charakter (např. GPS bod nebo Stát), jiné představují soubor elementárních vlastností (např. Způsob signalizace).



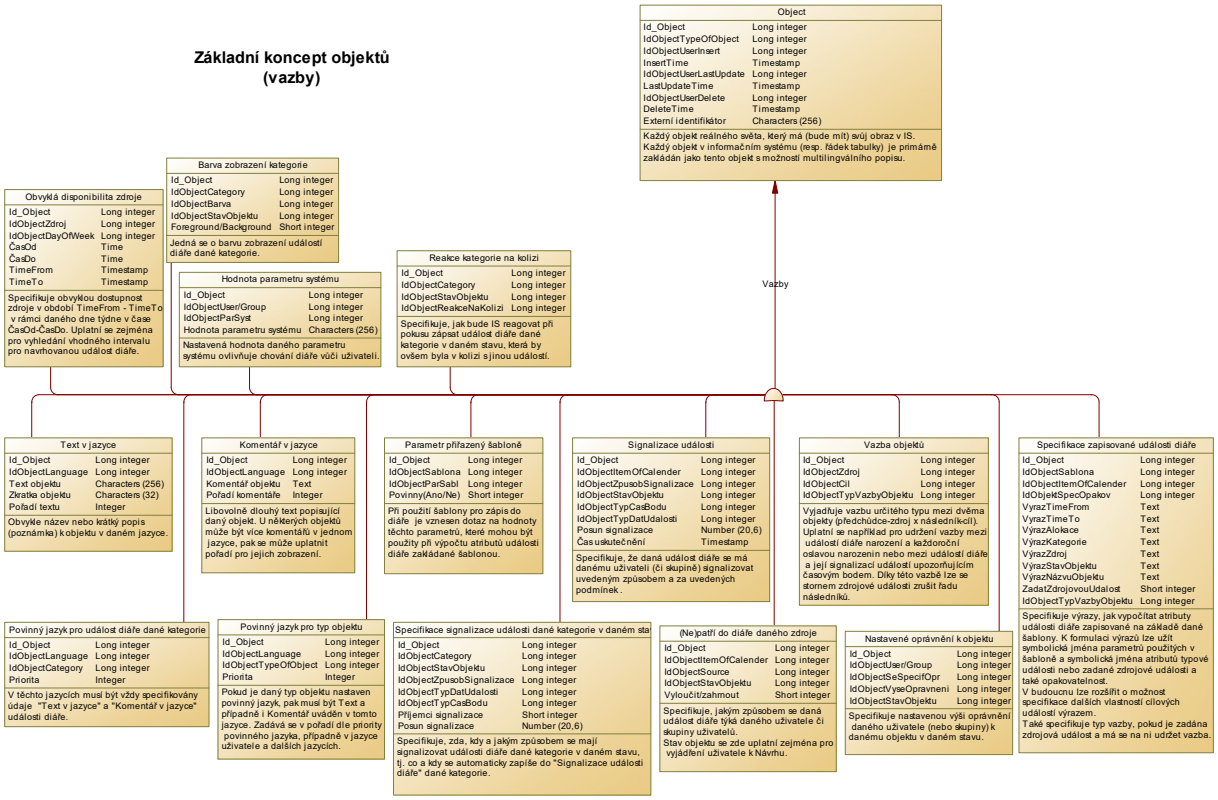
Obrázek 8- Základní koncept objektů (uživatelsky definované číselníky)

V části diagramu s Inheritancí „Číselníky nastavené programátory“ se prezentuje příslušnost k rodičovské entitě „Object“ (Parent) všech navržených entit (Children) tohoto typu.



Obrázek 9- Číselníky nastavené programátory

V části diagramu s Inheritancí „Vazby“ se prezentuje příslušnost k rodičovské entitě „Object“ (Parent) všech navržených entit (Children) tohoto typu.

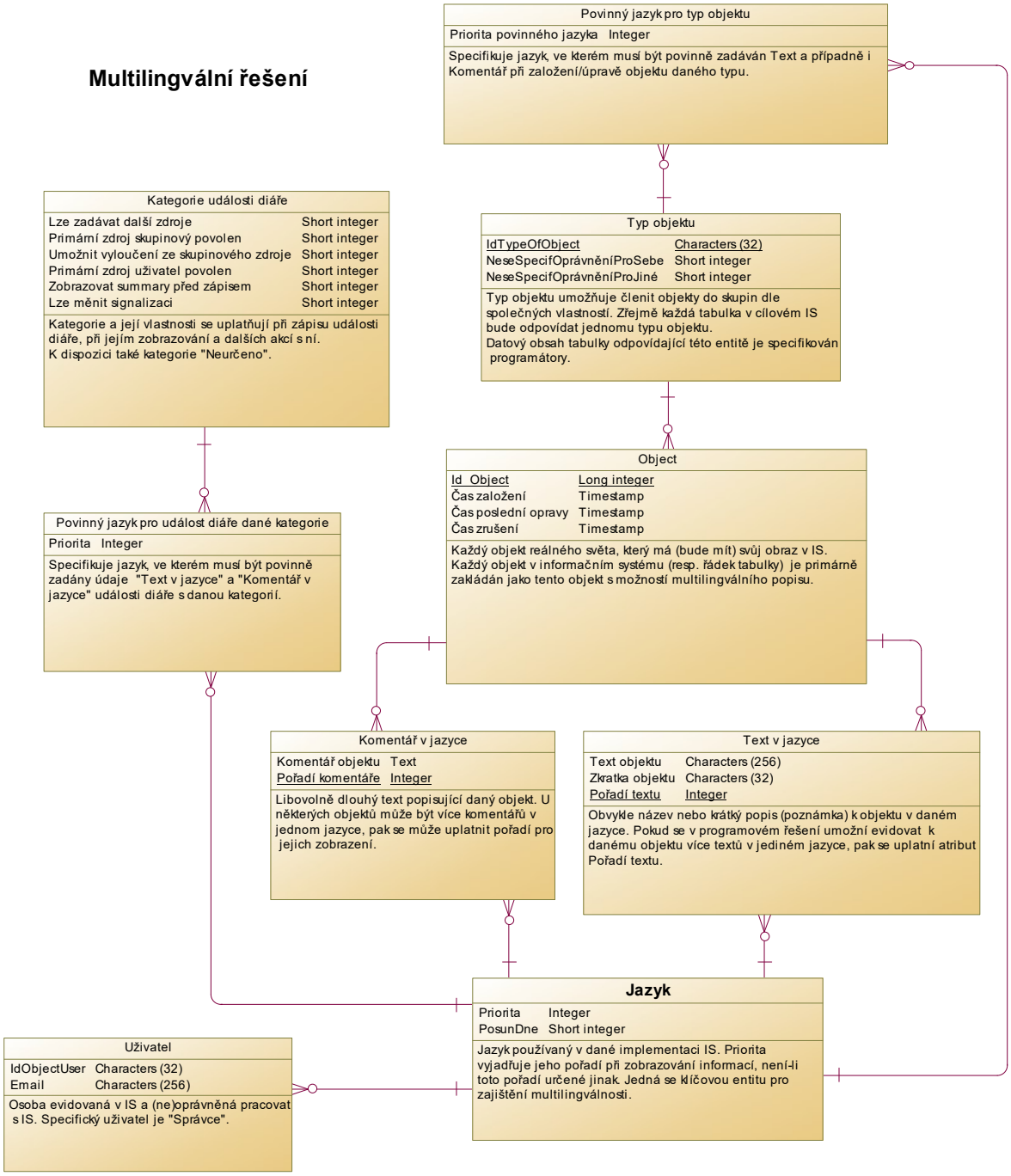


Obrázek 10- Základní koncept objektů (vazby)

5.2.6 Diagram CDM - Multilingválí řešení

Jádrem multilingválního řešení je entita „Jazyk“, jejíž záznamy jsou objektem jako vše v navrhovaném systému. Každý záznam v této entitě představuje jazyk (řeč), který je možné v systému používat. Atribut „Priorita“ se uplatní při stanovení jazyka pro zobrazení údajů. Každý uživatel systému (entita „Uživatel“) má přiřazen svůj implicitní jazyk.

Pro každý záznam v systému, tedy objekt, lze evidovat v entitě „Text v jazyce“ jeho název (resp. krátký popis či poznámka) v atributu „Text objektu“, jeho zkratku v atributu „Zkratka objektu“ v každém používaném (evidovaném) jazyce. Tyto údaje se mohou pro jeden objekt opakovat vícekrát (např. jednou jako název objektu, podruhé jako poznámka k objektu, …), pak atribut „Pořadí textu“ určuje jeho význam.



Obrázek 11- Multilingvální řešení

Pro každý záznam v systému, tedy objekt, lze evidovat v entitě „Komentář v jazyce“ komentář k němu v atributu „Komentář objektu“ v každém používaném (evidovaném) jazyce. Komentář může být jednomu objektu přiřazen opakovaně (např. komentář uživatele A, komentář uživatele B, …), pak atribut “Pořadí komentáře“ umožnuje jejich odlišení.

Entita „Povinný jazyk pro typ objektu“ umožnuje, aby pro daný typ objektu (např. alokace, událost diáře, …) byl specifikován povinný jazyk. To znamená, že zadávané textové údaje (entity „Text v

jazyce“ nebo „Komentář v jazyce“) daného typu objektu musí být uvedeny v povinném jazyce. Atribut „Priorita“ specifikuje, v jakém pořadí jazyků budou textové údaje zadávány.

V případě typu objektu „Událost diáře“ lze povinný jazyk specifikovat pro jednotlivé kategorie diáře v entitě „Povinný jazyk pro událost diáře dané kategorie“. Je-li pro danou kategorii zde specifikován, pak se v událostech diáře s danou kategorií neuplatní nastavení povinného jazyka pro typ objektu „Událost diáře“ specifikované v entitě „Povinný jazyk pro typ objektu“.

5.2.7 Diagram CMD – Zdroje diáře

Hlavní entitou tohoto diagramu je „Zdroj diáře“. Diagram se zaměřuje na specifikaci vlastností zdroje diáře, na členění zdrojů diáře a na vyjádření vztahů mezi uživatelem a skupinou uživatelů (jejich hierarchii).

Zdroj diáře reprezentuje objekt reálného světa, ke kterému se vztahují události diáře. Jinými slovy: Ke každému zdroji je veden jeho diář. Může se jednat o uživatele (konkrétní osoba s přístupem k IS), skupinu uživatelů anebo o prostředek (například auto, jednací místnost, …).

Programátory definovaný číselník, entita „Den týdne“, představuje jednotlivé dny v týdnu (Po, Út, St, Čt, Pá, So, Ne) a specifikuje jejich pořadí v rámci týdne (to může být v entitě „Jazyk“ posunuto, např. pro anglický jazyk). Data odpovídající entitě „Den týdne“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému.

Vazební entita „Obvyklá disponibilita zdroje“ specifikuje, kdy (hodina od – do) je daný zdroj obvykle disponibilní dle jednotlivých dnů týdne v daném období (disponibilita se totiž může v čase měnit, např. v zemědělství mohou mít v létě 9 hodinovou pracovní dobu a v zimě jen 7 hodinovou). Uplatní se zejména pro vyhledání vhodného volného intervalu pro navrhovanou událost diáře.

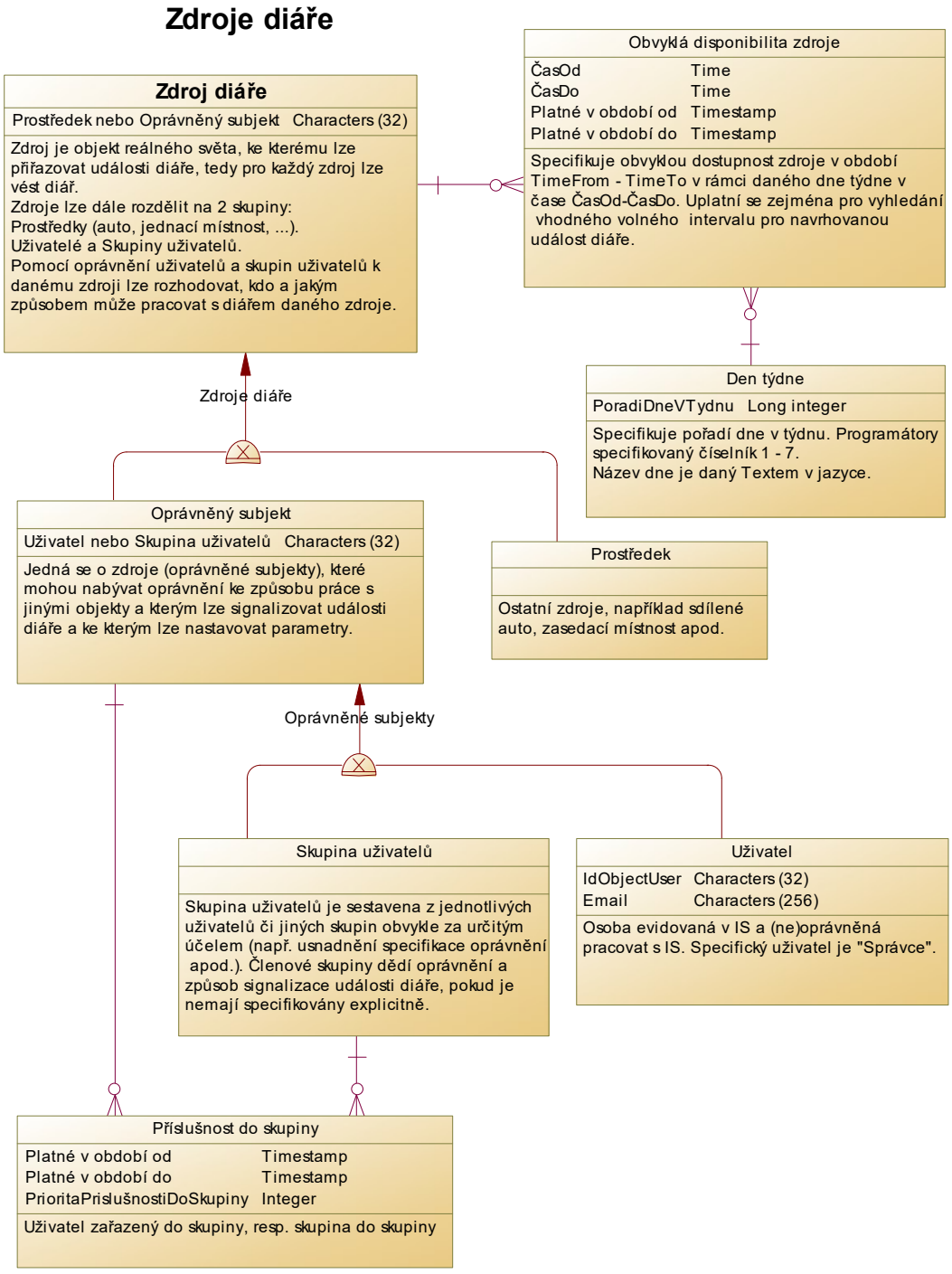
Zdroje lze primárně na dvě skupiny:

- Uživatel a skupina uživatelů (entita „Uživatel, Skupina uživatelů“) – pro tyto zdroje lze specifikovat oprávnění k jiným objektům, signalizaci událostí diáře, nastavení parametrů systému apod.
- Prostředek (entita „Prostředek“) – to jsou ostatní zdroje. Prostředek tím je myšleno zdroj, který může být např. firemní auto, zasedací místnost nebo jiná sdílená věc v rámci firmy.

Dále lze zdroje „Uživatel a skupina uživatelů“ rozdělit na:

- Uživatele (entita „Uživatel“) - osoba evidovaná v IS a (ne)oprávněná pracovat s IS. Uživatel, který má v rodičovské entitě „Objekt“ vyplněn „Čas zrušení“, již nemá přístup k IS a veškerá jeho nastavení nejsou účinná. Mezi vlastnosti uživatele patří jeho email. Specifický uživatel je "Správce", který je založen při instalaci systému.

- Skupiny uživatelů (entita „Skupina uživatelů“) - tím je myšlena skupina uživatelů, kteří využívají IS. Skupiny uživatelů usnadňují definice a stanovení oprávnění, signalizace událostí diáře, nastavení parametrů systému apod. Specifickou skupinou uživatelů je „Každý uživatel systému“, která je založena při instalaci systému a které jsou přiřazeny parametry systému. Každý nově založený uživatel je automaticky zařazen do této skupiny.



Obrázek 12- Zdroje diáře

Vazební entita „Příslušnost do skupiny“ specifikuje členy skupiny. Členem skupiny může být uživatel nebo skupina uživatelů. Jeden uživatel (jedna skupina uživatelů) může být členem libovolného počtu skupin.

Z příslušnosti do skupin mohou vyplývat oprávnění nebo určitá nastavení. Má-li uživatel specifikované nastavení přímo pro sebe, pak se skupinové nastavení nedědí (to fakticky umožní uživatele z dané skupiny vyloučit pro konkrétní situaci). Je-li uživatel členem více skupin, pak v situaci, kdy je specifikované oprávnění v těchto skupinách různé úrovně, dědí oprávnění nejvyšší. Je-li uživatel členem více skupin, pak v situaci, kdy je určité nastavení (nikoliv oprávnění) v těchto skupinách různé, dědí nastavení z té skupiny, do které je přiřazen s nejvyšší prioritou.

5.2.8 Diagram CDM – Alokace

Alokace je určena k tomu, aby se vědělo, kde se daná událost diáře odehrává. Tj. abychom věděli, kam máme jet, kolik si rezervovat času před zahájením události diáře, jaké dopravní prostředky si zajistit apod. Obecně alokace diáře specifikuje umístění objektů na Zemi.

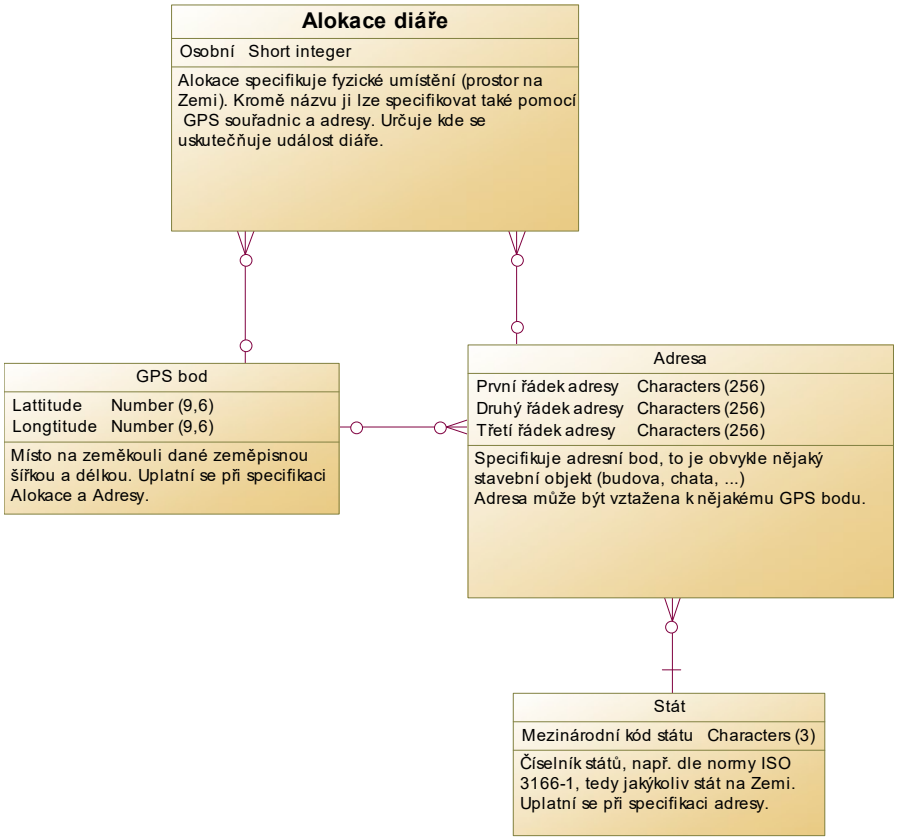
Alokaci, neboli umístění, je možné v tomto návrhu specifikovat volným textem anebo GPS souřadnicemi anebo adresou anebo jejich kombinací. Pokud je nastavena na osobní, může jí užívat pouze uživatel, který ji založil (to se uplatní například v situaci, kdy si do svého diáře plánuji rande se svou milenkou a nechci, aby o tom věděli kolegové).

Pokud je pro specifikaci alokace použity GPS souřadnice, uplatní se entita GPS bod, ve které se tyto souřadnice specifikují. Jeden GPS bod (souřadnice) se může použít pro více alokací. GPS bod může též zpřesňovat adresy.

Pokud je pro specifikaci alokace použita adresa, uplatní se entita Adresa. Adresa se specifikuje pomocí tří řádků adresy (ty se reálně píší na obálku) a státu. Tato struktura odpovídá standardům pro psaní adres minimálně v ČR. Někdy se adresa uvádí jen na dva řádky. Třetí řádek adresy v ČR je vždy PSČ a název pošty. První dva řádky se skládají dle určitých pravidel z těchto prvků: ulice, číslo popisné, číslo orientační, obec, městská část a část obce.

Stát představuje uživatelsky definovatelný číselník. Multilingvální pojetí umožňuje v entitě Text specifikovat jeho název v různých jazycích. Jedním z atributů entity Stát je mezinárodní kód státu.

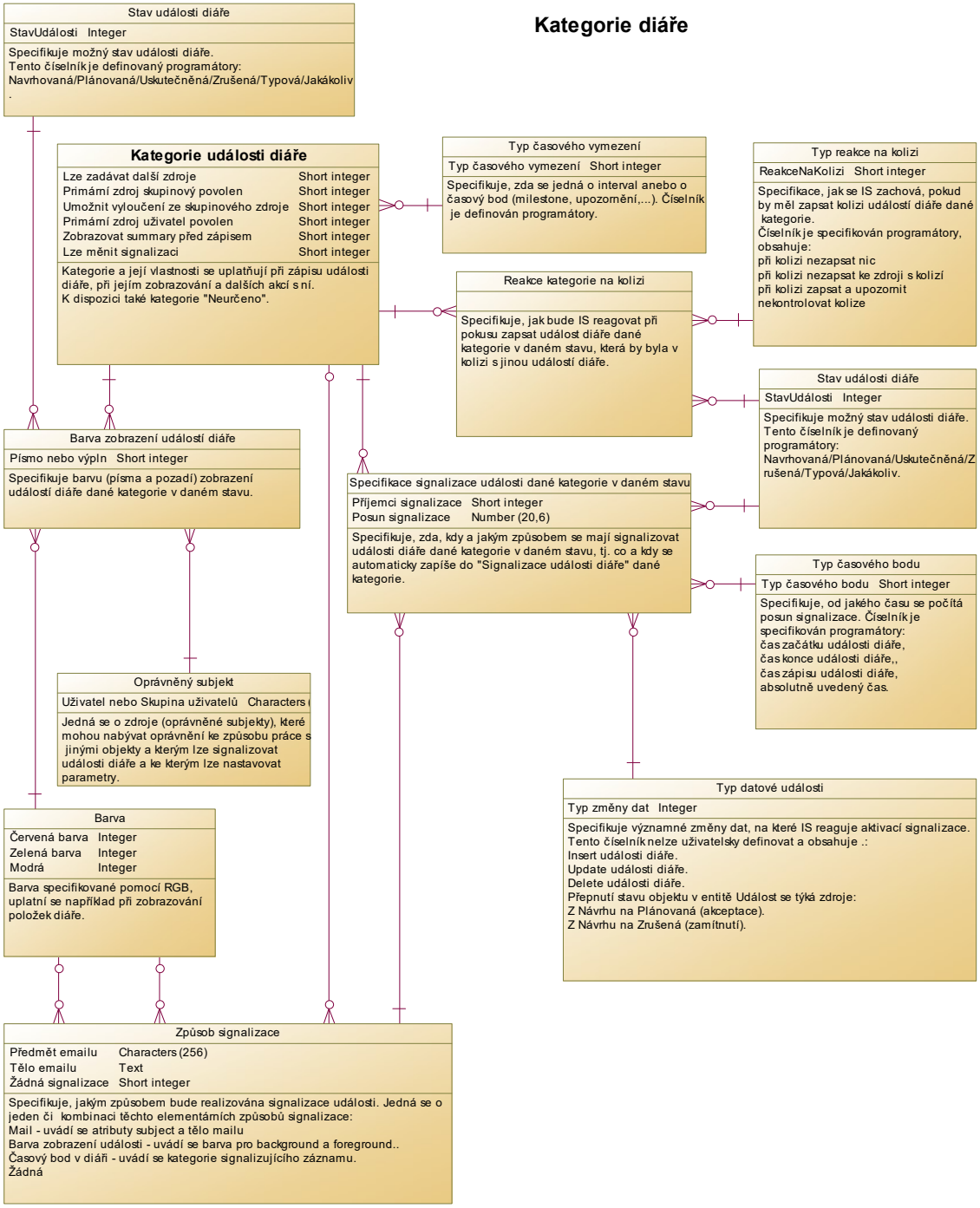
Pokud je pro specifikaci alokace použit volný text, tak je díky multilingválnímu řešení umístěn standardně v entitě Text.



Obrázek 13- Alokace diáře

5.2.9 Diagram CDM – Kategorie diáře

Hlavní (silnou) entitou v tomto diagramu představuje entita „Kategorie události diáře“. Kategorie diáře umožňuje přiřazovat událostem diáře určitý význam (např. Obchodní jednání, Práce na daném projektu, Práce v rámci daného firemního procesu, Úkol, Cíl, Narozeniny, …). Dále umožní jejich třídění, případné statistické vyhodnocování, a uplatní se také při uživatelském výběru podmnožiny událostí diáře či jejich statistickém hodnocení. Každá událost diáře je právě jedné kategorie.



Obrázek 14- Kategorie diáře

Tato kategorie poskytuje událostem diáře (přenáší do události) řadu vlastností (zejména barvu zobrazení, řešení kolizí a signalizaci), které se uplatní při založení události diáře nebo v jejím dalším životě (zobrazení, změna stavu, …).

Mezi přímé atributy kategorie, které ovlivňují vlastnosti události diáře s danou kategorií, patří:

- Lze zadávat další zdroje – je-li nastaveno na „Ano“, pak je možné k dané události diáře přiřadit více zdrojů, zřejmě k tomu bude uživatel i vyzván.
- Primární zdroj uživatel povolen – je-li nastaveno na „Ano“, pak je možné k dané události diáře při jejím založení přiřadit jako první zdroj uživatele.
- Primární zdroj skupinový povolen – je-li nastaveno na „Ano“, pak je možné k dané události diáře při jejím založení přiřadit jako první zdroj skupinu uživatelů.
- Umožnit vyloučení ze skupinového zdroje - je-li nastaveno na „Ano“, pak je možné u dané události diáře specifikovat, že se týká dalšího uživatele, tj. další zdroj, který bude mít stav události nastaven na „Zrušená“. Třebaže je daný uživatel členem skupiny uživatelů, jejíž členové mají danou událost diáře navrženou k akceptování, stejně se ho tato událost diáře netýká (nebude se mu zobrazovat ani signalizovat).
- Zobrazovat sumář před zápisem – je-li nastaveno na „Ano“, pak se před zápisem dané události diáře zobrazí komplexní informace o ní, zejména koho a jak se týká, komu a jak bude signalizována. Uživatel se pak může rozhodnout, zda je to OK anebo se to musí upravit.
- Lze měnit signalizaci – je-li nastaveno na „Ano“, pak je možné před zápisem dané události diáře změnit její signalizaci, která se připravila automaticky.
- Automaticky posunout časový bod – je-li nastaveno na „Ano“, pak se časový bod (událost diáře dané kategorie – například úkol) automaticky přesune z uplynulého dne na aktuální den, pokud je ve stavu „Navržená“ nebo „Plánovaná“. Zajistí, že se nesplněné úkoly nečinností neztratí, ale že je nutné je splnit anebo vědomě zrušit.

Současně je v tomto diagramu zobrazena řada číselníků definovaných programátory i uživatelsky, které slouží pro nastavení standardizovaných vlastností kategorií a zprostředkovaně i událostí diáře.

Programátory definovaný číselník „Typ časového vymezení“ určuje, jak se událost diáře s danou kategorií vymezuje vůči času. Data odpovídající entitě „Typ časového vymezení“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému. Jedná se tyto typy časového vymezení:

- Časový interval – v události diáře se specifikuje časem zahájení a odlišným časem ukončení.
- Časový bod – v události se specifikuje shodným časem zahájení i ukončení, případně se čas ukončení neuvádí.

Časové vymezení také ovlivňuje uživatelský vstup dat, např. v případě zadání časového bodu uživatel neuvádí čas ukončení události diáře. Data v tabulkách budoucího IS odpovídající této entitě musí být automaticky vytvořena při instalaci systému.

Uživatelsky definovatelný číselník s barvami (entita „Barva“) umožňuje definovat barvy, které se následně použijí při zobrazení událostí diáře níže uvedeným způsobem. Barvy se specifikují pomocí barevných složek RGB.

Vazební entita „Barva zobrazení kategorie“ umožňuje nastavit, aby se událost diáře s danou kategorií zobrazovala zde uvedenou barvou písma. Barvu lze nastavit individuálně pro každého uživatele či skupinu uživatelů, a to ještě v závislosti na stavu události diáře. Nastavení barvy písma (Foreground) a barvy výplně (Background) se vždy specifikuje samostatným záznamem v entitě „Barva zobrazení kategorie“. Zde nastavené barvy zobrazení mají vyšší prioritu než barvy uvedené v parametrech systému, ale nižší prioritu než případná signalizace realizovaná barvou zobrazení v diáři.

Vazební entita „Reakce kategorie na kolizi“ umožňuje specifikovat, jak se má budoucí IS chovat při pokusu zapsat událost diáře dané kategorie v daném stavu, která by byla v kolizi s jinou událostí diáře v daném stavu. Pokud mají události v kolizi nastavenu různou reakci, zvolí se ta mírnější, tj. ta která umožní zápis.

Možnosti reakce na kolizi událostí diáře jsou definovány programátory specifikovaným číselníkem v entitě „Typ reakce na kolizi“. Data odpovídající entitě „Typ reakce na kolizi“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému. Jedná se tyto typy reakce:

- Při kolizi nezapsat nic – při identifikaci kolize (alespoň jedním zdrojem) při zápisu události diáře se událost nezapíše, tedy ani nepřiřadí zdrojům, které by v kolizi nebyly.
- Při kolizi nezapsat ke zdroji s kolizí – při identifikaci kolize při zápisu události diáře se událost zapíše a přiřadí se jen ke zdrojům, které v kolizi nejsou; pokud jsou všechny zdroje v kolizi, pak se nezapíše ani událost diáře.
- Při kolizi zapsat a upozornit – při identifikaci kolize při zápisu události diáře se událost zapíše a přiřadí ke všem relevantním zdrojům a současně je o kolizi obsluha informována.
- Nekontrolovat kolize – při identifikaci kolize při zápisu události diáře se událost zapíše a přiřadí ke všem relevantním zdrojům; toto se uplatní například pro časové body (narozeniny apod.).

Uživatelsky definovatelný číselník, entita „Způsob signalizace“, umožňuje uživatelům specifikovat signalizaci, tj. upozornění na významné události v diáři, těmito elementárními způsoby (případně jejich kombinací) :

- Barvou zobrazení události v diáři – v takovém případě se zde uvádí barva písma (Foreground) a barva výplně (Backround) z číselníku barev; události diáře se signalizací tímto způsobem se pak zobrazují v těchto (obvykle výrazných) barvách.

- Upozorňovacím časovým bodem – v takovém případě se zde uvádí kategorie události diáře, kterou ponese upozorňovací bod realizovaný událostí diáře. Tento bod (obvykle výrazné barvy, např. kategorie „Zítra je důležitá událost“) se pak prezentuje v diáři relevantních uživatelů.
- Odesláním mailu – v takovém případě se zde uvádí znění subjektu a těla upozorňovacího mailu. Odpovídající mail pak upozorňuje relevantní uživatele na významnou událost diáře.
- Žádná signalizace – v takovém případě se zde uvádí, že se nemá signalizovat. To se obvykle uplatní v situaci, kdy se určitá událost diáře signalizuje nějaké skupině uživatelů, ale je potřebné určitého uživatele ze signalizace vyloučit.

Programátory definovaný číselník, entita „Typ datové události“, představuje situace (změny v datech), které aktivují signalizaci události diáře (atribut „Čas aktivace“ v entitě „Signalizace události“) uživatelům. Uplatní se v entitě „Specifikace signalizace události dané kategorie v daném stavu“. Data odpovídající entitě „Typ datové události“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému. Jedná se tyto datové události aktivující signalizaci:

- Založení události diáře (insert) – při vložení události diáře.
- Oprava události diáře (update) – při opravě události diáře.
    - Přepnutí z „navrhovaná“ na „plánovaná“.
    - Přepnutí z „navrhovaná“ na „zamítnutá“ (tj. zrušení události diáře).
    - …
    - Jakákoliv.
- Uživatelova akceptace návrhu – při přepnutí stavu události v diáři daného uživatele (entita „Událost se týká zdroje“) z „navrhovaná“ na „plánovaná“.
- Uživatelovo zamítnutí návrhu – při přepnutí stavu události v diáři daného uživatele (entita „Událost se týká zdroje“) z „navrhovaná“ na „zamítnutá“.

Programátory definovaný číselník, entita „Typ časového bodu“, představuje typ výchozího času, od kterého je možné odvodit čas signalizace. Uplatní se v entitě „Specifikace signalizace události dané kategorie v daném stavu“. Data odpovídající entitě „Typ časového bodu“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému. Jedná se tyto typy časového bodu:

- Čas zahájení události diáře.
- Čas ukončení události diáře.
- Čas zápisu události diáře.
- Čas opravy události diáře.
- Čas vyjádření k návrhu (zápis do entity „Událost se týká zdroje“).
- Absolutně uvedený čas.

Vazební entita „Specifikace signalizace události dané kategorie v daném stavu“ specifikuje, zda a jak mají být signalizovány události diáře uvedené kategorie. Je-li uvedena, pak se v okamžiku založení události diáře dané kategorie, založí i odpovídající signalizace. Specifikuje se uvedením:

- Stav události diáře – signalizace se uplatní, pokud je událost diáře ve zde uvedeném stavu.
- Příjemci signalizace – buď:
    - Všichni uživatelé, kteří mají danou událost přiřazenu do svého diáře.
    - Všichni uživatelé, kteří mají danou událost přiřazenu do svého diáře, kromě toho uživatele, který danou událost zapsal (autor).
    - Jen ten uživatel, který danou událost zapsal (autor).
- Způsob signalizace – viz entita „Způsob signalizace“.
- Typ datové události – kdy se aktivuje signalizace (viz entita „Typ datové události“), to ale neznamená, že se hned uskuteční.
- Typ časového bodu – určuje, kdy se má uskutečnit signalizace, resp. od kdy se bude počítat čas pro uskutečnění signalizace.
- Časový posun – uvádí v hodinách, jak se má posunout uskutečnění signalizace od uvedeného výchozího času specifikovaného Typem časového bodu.

Příklad nastavení signalizace pro kategorii události diáře „Schůzka řešitelů projektu“. První signalizace (informace o návrhu) bude nastavena takto:

- Stav události „Navrhovaná“.
- Příjemci „Všichni kromě autora události“.
- Způsob signalizace „Email“ s textem „Prosím přijměte návrh v diáři“.
- Typ datové události „Čas zápisu události diáře“.
- Typ časového bodu „Čas zápisu události diáře“.
- Časový posun = 0.

Autor založí událost ve stavu „Navrhovaná“ s danou kategorií a specifikuje jako zdroj skupinu uživatelů představující projektový tým „Projektový tým 1“. V tomto případě dojde k okamžitému odeslání mailů všem, kteří se mají dané schůzky účastnit, tj. členům projektového týmu „Projektový tým 1“.

Druhá signalizace (informace o zamítnutí návrhu) je nastavena takto:

- Stav události „Navrhovaná“.
- Příjemci „Autor události“.
- Způsob signalizace „Email“ s textem „Zamítám Váš návrh v diáři …“.
- Typ datové události „Uživatelovo zamítnutí návrhu“.

- Typ časového bodu „Čas zápisu události diáře“.
- Časový posun = 0.

Uživatel se vyjádřil k návrhu zamítavě, tj. do entity „Událost se týká zdroje“ byl vložen nový záznam vztažený k dané události a danému uživateli se stavem události diáře „Zrušená“. V tomto případě dojde k okamžitému odeslání mailu navrhovateli schůzky.

Třetí signalizace (informace o akceptaci návrhu) je nastavena takto:

- Stav události „Navrhovaná“.
- Příjemci „Autor události“.
- Způsob signalizace „Email“ s textem „Akceptuji Váš návrh v diáři …“.
- Typ datové události „Uživatelovo akceptování návrhu“.
- Typ časového bodu „Čas zápisu události diáře“.
- Časový posun = 0.

Uživatel se vyjádřil k návrhu zamítavě, tj. do entity „Událost se týká zdroje“ byl vložen nový záznam vztažený k dané události diáře a k danému uživateli se stavem události diáře „Plánovaná“. V tomto případě dojde k okamžitému odeslání mailu navrhovateli schůzky. Toto zopakují i zbývající uživatelé, členové „Projektový tým 1“.

Čtvrtá signalizace je nastavena takto:

- Stav události „Plánovaná“.
- Příjemci „Všichni uživatelé“.
- Způsob signalizace „Časovým bodem“ s kategorií „Připomínka zítřejší schůzky“.
- Typ datové události „Přepnutí události diáře z navrhovaná na plánovaná“.
- Typ časového bodu „Čas zahájení události diáře“.
- Časový posun = -24.

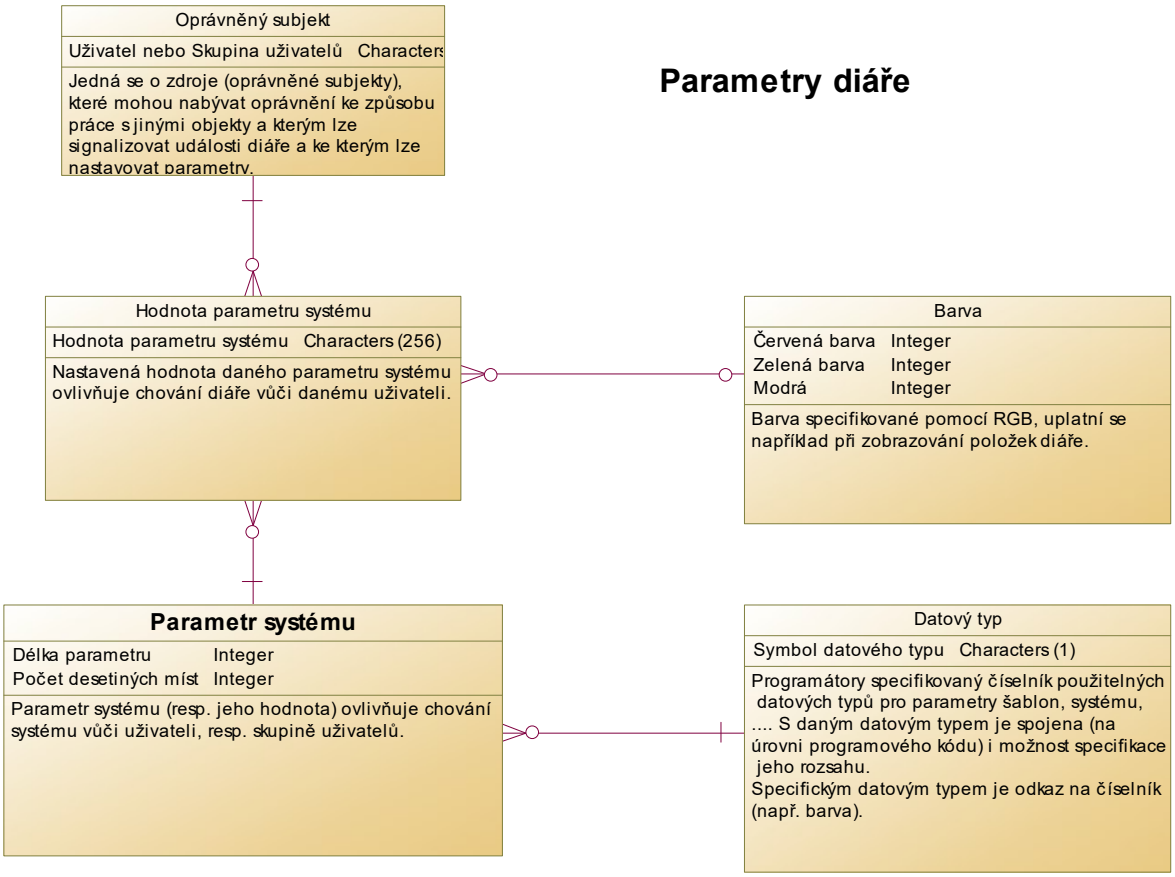
Autor vyhodnotil vyjádření členů skupiny a rozhodl se schůzku uskutečnit i bez přítomnosti jednoho člena, tedy změnil její stav na „Plánovaná“, čímž se aktivovala uvedená signalizace, která bude mít účinnost až 24 hodin před zahájením schůzky.

Pokud by se rozhodl schůzku zrušit, tak by se uplatnila například takováto signalizace pro odeslání informačního mailu:

- Stav události „Zrušená“.
- Příjemci „Všichni kromě autora události“.
- Způsob signalizace „Mail“ s textem „Ruším svůj návrh v diáři …“.
- Typ datové události „Přepnutí události diáře z navrhovaná na zrušená“.
- Typ časového bodu „Čas poslední opravy diáře“.
- Časový posun = 0.

5.2.10 Diagram CDM – Parametry diáře

Hlavní (silnou) entitou v tomto diagramu představuje entita „Parametr systému“. Parametr systému (resp. jeho hodnota) ovlivňuje chování systému vůči uživateli, resp. skupině uživatelů.



Obrázek 15- Parametry diáře

Jedná se o číselník definovaný programátory, takže data odpovídající entitě „Parametr systému“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému. Obsahuje:

- Dny do minulosti – kolik dnů do minulosti od aktuálního data se má implicitně zobrazit při spuštění diáře, je přednastaveno na jeden den.
- Dny do budoucnosti – kolik dnů do budoucnosti od aktuálního data se má implicitně zobrazit při spuštění diáře, je přednastaveno na sedm dní.
- Implicitní barva písma (Foreground) zobrazovaných událostí diáře. Tato barva má nižší prioritu než barva specifikovaná pro určitou kategorii diáře. Je přednastavena černá barva.
- Implicitní barva výplně (Background) zobrazovaných událostí diáře. Tato barva má nižší prioritu než barva specifikovaná pro určitou kategorii diáře. Je přednastavena bílá barva.
- Způsob písma pro události ve stavu „Navržená“ je přednastaven na „kurziva“.
- Způsob písma pro události ve stavu „Plánovaná“ je přednastaven na „běžné“.
- Způsob písma pro události ve stavu „Uskutečněná“ je přednastaven na „přeškrtnuté“.

- Způsob písma pro události v kolizi je přednastaven na „tučné“.
- Implicitní čas dne, ve kterém se zobrazí časový bod, který nemá explicitně uveden čas (je specifikován jen datem).
- Zda automaticky přepnout stav v entitě „Událost se týká zdroje“ na „Zrušená“, pokud se tento stav nastaví v entitě „Událost diáře“. Přednastaveno na „Ano“ (v opačném případě si musí každý uživatel přepnout sám).
- Zda automaticky přepnout stav v entitě „Událost diáře“ na „Plánovaná“, pokud si ji všichni relevantní uživatelé přepnuli do stavu „Plánovaná“.
- Perioda, ve které se bude vyhodnocovat signalizace. Tj. jak často se má aktivovat (zřejmě databázový event) pro uskutečnění signalizace (mail, signalizační bod).
- Čas dne, ve kterém se budou vyhodnocovat události diáře (časové body – úkoly a cíle), aby se automaticky posunulo jejich datum o jeden den. Pokud je tento čas mezi 0:00 až 12:00, tak se posouvá na aktuální datum, jinak na další den.

Programátory definovaný číselník, entita „Datový typ“, specifikuje datový typ parametrů systému, také se uplatní při uživatelské specifikaci parametrů šablony. Jeho obsah bude odpovídat standardům použitého programovacího jazyka. Případně může být datový typ specifikován odkazem na číselník (např. barvu), pak se uživatelské zadání hodnoty tohoto parametru uskuteční formou překryvného seznamu.

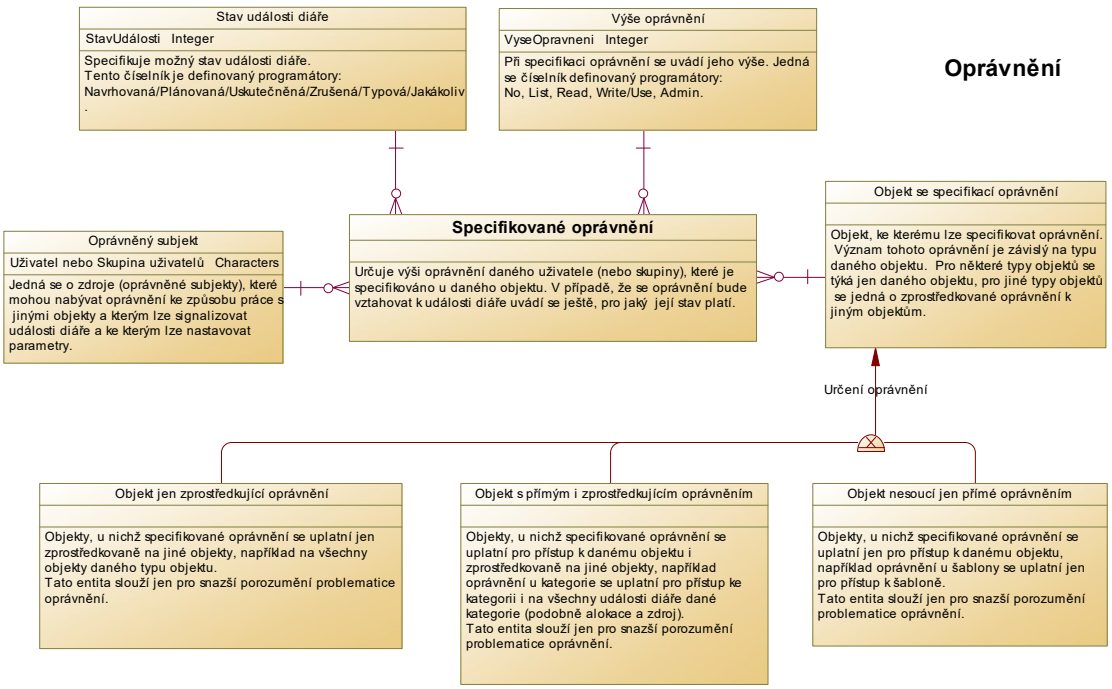
Vazební entita „Hodnota parametru systému“ specifikuje nastavení daného parametru systému pro daného uživatele či skupinu uživatelů. Některé hodnoty mohou odkazovat na číselník (např. barvu). V rámci instalace budou tyto hodnoty parametrů systému naplněny (viz výše) pro skupinu uživatelů „Každý uživatel systému“, tedy zprostředkovaně pro každého nového uživatele.

5.2.11 Diagram CDM - Specifikace oprávnění

K objektům některých typů lze specifikovat oprávnění, tedy jsou nositelem specifikace oprávnění. V rámci "Multilingválního diáře" se jedná o tyto typy objektů: Alokace diáře, Zdroj diáře, Kategorie události diáře, Šablona diáře a Typ objektu.

Objekty se specifikací oprávnění (nositelé oprávnění), tedy i specifikovaná oprávnění, můžeme rozdělit na tři skupiny dle toho, čeho se uvedené oprávnění týká. V diagramu jsou tyto skupiny reprezentovány těmito entitami (tyto entity jsou v diagramu uvedeny jen z důvodu názornosti pro pochopení problematiky oprávnění a nepředpokládám jejich projekci do fyzického datového modelu či databáze):

- Objekt jen zprostředkující oprávnění.
- Objekt s přímým i zprostředkujícím oprávněním.
- Objekt nesoucí jen přímé oprávněním.



Obrázek 16- Oprávnění

Do první skupiny patří objekty z entity "Typ objektu". Oprávnění rozhoduje o zprostředkovaném oprávnění ke všem objektům daného typu. Např. oprávnění u typu objektu "Barva" ovlivní zprostředkovaně, zda daný uživatel může barvy prohlížet v seznamu či v detailu, zakládat nebo i spravovat, apod.

Do druhé skupiny patří objekty z entit "Zdroj diáře", "Kategorie události diáře" a "Alokace diáře". Oprávnění rozhoduje o přímém oprávnění k objektu (nositeli oprávnění) i zprostředkovaném

oprávnění k událostem diáře. Např.: V případě objektu se specifikací oprávnění “Automobil 2C4 5016“ (typ objektu “Prostředek“) oprávnění určuje, zda daný uživatel může daný objekt (auto) prohlížet v seznamu či v detailu anebo i administrovat. Současně oprávnění zprostředkovaně určuje, zda daný uživatel může události diáře s daným zdrojem (autem) vidět, vidět v detailu anebo i zapisovat.

Do třetí skupiny patří objekty z entity “Šablona diáře“. Oprávnění rozhoduje o přímém oprávnění k objektu (nositeli oprávnění). Např. oprávnění u objektu “Pravidelná týdenní porada projektového týmu“ (typ objektu “Šablona diáře“) přímo rozhodne, zda daný uživatel může daný objekt (šablonu pro týdenní poradu) prohlížet v seznamu či v detailu, použít pro generování událostí diáře nebo i spravovat.

Programátory definovaný číselník, entita “Výše oprávnění“, specifikuje možné úrovně oprávnění. Data odpovídající entitě „Výše oprávnění“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému. Jedná se tyto výše oprávnění:

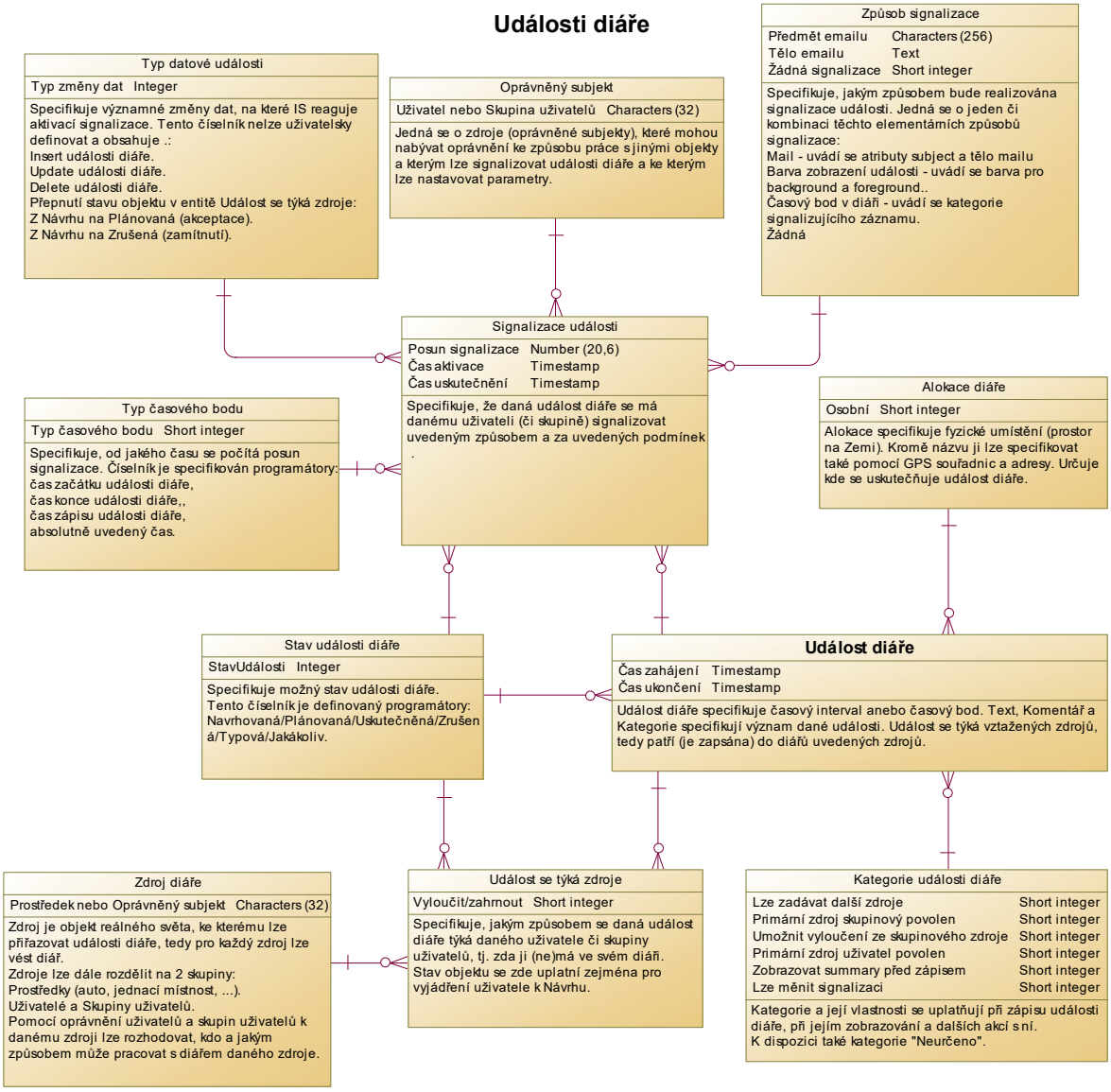
- No: Žádný přístup.
- List: Čtení jen dílčích údajů (např. od-do v diáři).
- Read: Čtení všech údajů, včetně detailních.
- Write/Use: Je-li nositelem oprávnění objekt typu „Typ objektu“, jedná se oprávnění objekty daného typu zakládat; není-li nositelem oprávnění objekt typu „Typ objektu“, jedná se oprávnění používat daný objekt (například použít šablonu pro generování událostí diáře, zapsat událost diáře dané kategorie apod.)
- Admin: Oprávnění objekt spravovat.

Vazební entita “Specifikované oprávnění“ stanovuje výši (úroveň) oprávnění pro daného uživatele anebo pro skupinu uživatelů. Toto oprávnění je vztaženo k nositeli oprávnění (entita „Objekt se specifikací oprávnění“). Pokud se oprávnění uplatňuje zprostředkovaně k události diáře, pak výše oprávnění ještě závisí na jejím stavu (entita „Stav události diáře“).

Aktuálně platná výše oprávnění daného uživatele se určí takto: Je-li oprávnění specifikováno přímo na jeho osobu (uživatele) použije se, jinak se použije nejvyšší oprávnění vyplývající ze členství ve skupinách.

5.2.12 Diagram CDM – Událost diáře

Hlavní (silnou) entitu v tomto diagramu představuje entita „Událost diáře“, které je fakticky stěžejní entitou celého tohoto návrhu. Událost diáře reprezentuje událost z reálného světa, kterou je vhodné díky její významnosti zaznamenat do diáře. Událost je v časové ose vymezena časovým intervalem anebo časový bodem (časové razítko – timestamp).



Obrázek 17- Události diáře

Každá událost diáře se nachází v určitém stavu, který se během jejího života může měnit (zřejmě nejčastěji takto: Navrhovaná -> Plánovaná -> Realizovaná). Stav události ovlivňuje její zobrazení v diáři, lze ho použít pro specifikaci signalizace, reakce na kolizi a přístupových práv. Možné stavy jsou definovány programátory nastaveným číselníkem, entita „Stav události diáře“. Data odpovídající entitě „Stav události diáře“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému. Jedná se tyto stavy:

- Navrhovaná – použije se obvykle tehdy, kdy autor události zapsal do diáře někoho jiného a očekává jeho odsouhlasení.
- Plánovaná – použije se obvykle tehdy, když autor neočekává/nepožaduje odsouhlasení nebo když navržená událost byla odsouhlasena.

- Uskutečněná – použije se obvykle tehdy, když se plánovaná událost uskutečnila s cílem dosáhnout větší přehlednosti zobrazení diáře, je-li to potřebné.
- Zrušená – použije se obvykle tehdy, když navržená událost je zamítnuta účastníky a neuskuteční se nebo když se plánovaná událost z nějakého důvodu v reálném světě neuskutečnila.
- Typová – lze použít jen pro specifikaci šablony, v diáři se standardně nezobrazuje.
- Jakákoliv – lze použít jen pro specifikaci oprávnění, reakcí na kolize a signalizace.

Význam (obsah, důvod evidování) události je určen zejména multilingválním popisem (viz entity „Text v jazyce“ a „Komentář v jazyce“) a dále přiřazením právě jedné kategorie (entita „Kategorie události diáře“).

Kategorie může dané události diáře dále zprostředkovat (viz též diagram Kategorie diáře):

- Barvu zobrazení – barva zobrazení události v diáři daného uživatele může být daná kategorií (viz entita „Barva zobrazení kategorie“ v CDM diagramu „Kategorie diáře“).
- Signalizaci – signalizace událostí diáře (entita „Signalizace události“) významných kategorií může být automaticky zakládána (viz entita „Specifikace signalizace události dané kategorie v daném stavu“ v CDM diagramu „Kategorie diáře“).
- Reakci na kolizi – reakce na kolizi může být specifikována odlišně pro každou kategorii (viz entita „Reakce kategorie na kolizi“ v CDM diagramu „Kategorie diáře“).
- Oprávnění – oprávnění (entita „Specifikované oprávnění“ v diagramu „Specifikace oprávnění“) uživatelů, resp. skupin uživatelů, specifikovaná u objektů typu „Kategorie události diáře“ se uplatní i pro přístup k událostem diáře dané kategorie.

Událost diáře se týká vztažených zdrojů, tedy patří (je zapsána) do diářů uvedených zdrojů (entita „Událost se týká zdroje“). Stav události diáře ve vztahu k danému zdroji (atribut „Stav události diáře“ v entitě „Událost se týká zdroje“) se může lišit oproti stavu události – to se obvykle uplatní, když se má uživatel vyjádřit (akceptovat či zamítnout) k návrhu události, která se ho týká. Tímto stavem se pro daného uživatele řídí zobrazení události v diáři, signalizace apod. Pokud se daná událost uživatele týká přímo (zdrojem je uživatel) i nepřímo (zdrojem je skupina uživatelů, do které patří), pak se uplatní jen ten záznam z entity „Událost se týká zdroje“, který se uživatele týká přímo.

Oprávnění (entita „Specifikované oprávnění“ v diagramu „Specifikace oprávnění“) uživatelů, resp. skupin uživatelů, specifikovaná u objektů typu „Zdroj“ se uplatní zprostředkovaně i pro přístup k událostem diáře tak, že uživatel si může zobrazit diáře jen těch zdrojů, ke kterým má oprávnění alespoň ve výši „List“, a to způsobem, který mu oprávnění umožňuje. Oprávnění k jedné události diáře se tedy může lišit v závislosti na tom, který zdroj byl pro zobrazení diáře zvolen.

Alokace, entita „Alokace diáře“ (viz diagram „Alokace“), specifikuje, kde se daná událost diáře odehrává (koná). Oprávnění (entita „Specifikované oprávnění“ v diagramu „Specifikace oprávnění“)

uživatelů, resp. skupin uživatelů, specifikovaná u objektů typu „Alokace diáře“ se uplatní i pro přístup k událostem diáře konané v dané alokaci.

Zde ještě uveďme, že finální výše oprávnění k dané události, je nejnižší z oprávnění specifikovaných u objektu:

- Typ objektu „Událost diáře“.
- Relevantní kategorie diáře.
- Relevantní alokace diáře.
- Relevantní zdroj diáře zvolený při zobrazení diáře.

Vazební entita „Signalizace události“ umožňuje, aby daná událost diáře, entita „Událost diáře“, byla signalizována danému uživateli (případně skupině uživatelů), entita „Uživatel, Skupina uživatelů“, uvedeným způsobem, entita „Způsob signalizace“, za uvedených podmínek, entity „Stav události diáře“ a „Typ časového bodu“ a „Typ datové události“.

Signalizace může být založena automaticky při založení události diáře, je-li definována pro kategorii v ní použitou (viz CDM diagram „Kategorie diáře“), anebo uživatelem. Aby byla signalizace aktivní, tj. aby se vyhodnocovala, tak musí mít uveden „Čas aktivace“, tedy musel nastat odpovídající typ datové události. Pokud je signalizace realizována, tak se čas uskutečnění uvede v atributu „Čas uskutečnění“ entity „Signalizace události“ (to neplatí pro signalizaci barvou zobrazení v diáři).

Příklad události diáře s uživatelem zadanou signalizací:

- Založím důležitou událost diáře (od 1.4.2017 16:00 do 1.4.2017 18:00) s kategorií „Osobní schůzka Martina Jandy“ (kromě mě mají všichni nastaveno oprávnění k této kategorii ve výši „List“ – tedy vidí, že mám uvedený čas naplánovaný, ale nevidí proč) ve svém diáři (zdrojem je uživatel „Martin Janda“) na alokaci „Neuvedeno“ ve stavu „Plánovaná“.
- Připojím signalizace (entita „Signalizace události“) pro stav „Plánovaná“ s typem časového bodu „Čas zahájení události diáře“ a s typem datové události „Založení události diáře“ (tedy se hned vyplní i „Čas aktivace“ časem založení události diáře):
    - Způsobem „Odeslání mailu“ s posunem -1 hodina.
    - Způsobem „Barva zobrazení v diáři“ s posunem -12 hodin.
    - Způsobem „Upozorňovací časový bod“ s posunem 26 hodin (kategorie „Upozornění“).
- V čase 31. 3. 2017 cca v 14:00 systém automaticky vygeneruje událost diáře kategorie „Upozornění“ (upozorňovací časový bod) – viz „BPM Signalizace“. Ta bude provázána s původní události diáře (viz entita „Vazba objektů” v diagram CDM “Koncept objektů”). Současně se vyplní atribut čas uskutečnění. Tímto se v mém zobrazeném diáři objeví silná červená čára, ze které se mohu pohodlně podívat na výchozí událost (např. po najetí myši). Pokud výchozí událost převedu do stavu „Uskutečněná“, upozornění se přestane zobrazovat.

- V čase 1. 4. 2017 ve 4:00 se začne výchozí událost (16:00 - 18:00) zobrazovat v mém diáři nějakou signální barvou, abych na ni celý den myslel. Pokud ji převedu do stavu „Uskutečněná“, zvýraznění se odstraní.
- V čase 1. 4. 2017 cca v 15:00 systém automaticky odešle mail a současně se vyplní atribut čas uskutečnění. Pokud bych před uvedeným časem převedl událost do stavu „Zrušená“, mail by se neodeslal.

Příklad události diáře s automaticky založenou signalizací (viz též diagram CDM „Kategorie diáře“):

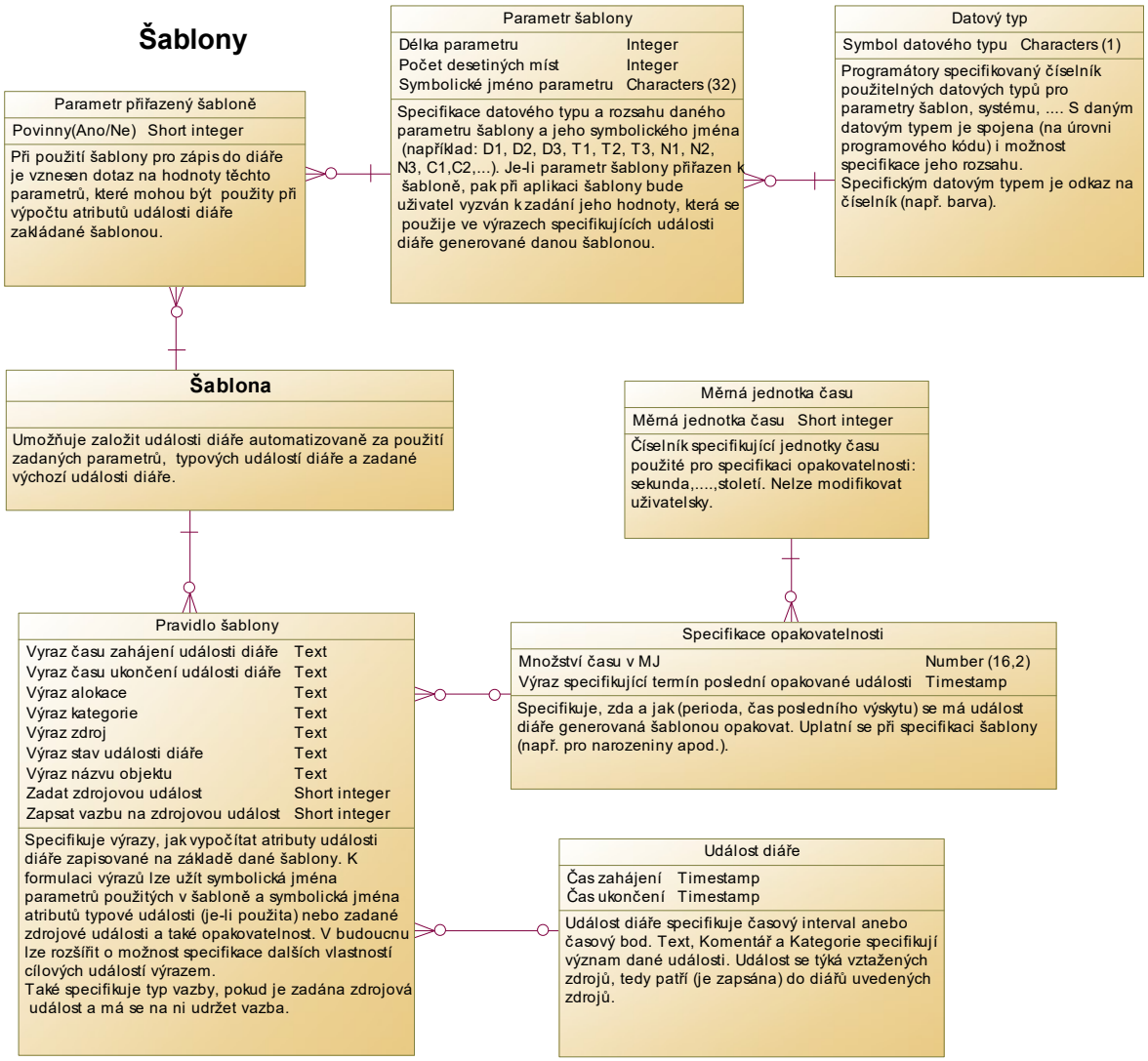
- Založím událost diáře (od 1.4.2017 14:00 do 1.4.2017 17:00) s kategorií „Schůzka projektového týmu“ se vztaženým zdrojem uživatelem (já „Martin Janda“) a zdrojem skupinou uživatelů „Projektový tým 1“ na alokaci „Zasedací místnost 1“ ve stavu „Navrhovaná“.
- Mimo jiné bude danou kategorií předdefinována signalizace (tedy se hned po založení události automaticky zapíše do entity „Signalizace události“) pro stav „Zrušená“ s typem časového bodu „Čas opravy události diáře“, s typem datové události „Přepnutí navrhované na zrušenou“, se způsobem „Odeslání mailu“ všem kromě autora a s posunem 20 minut.
- Následně zjistím, že mi tato událost koliduje s novou důležitější událostí diáře, a dne 28. 3. 2017 v 12:00 schůzku projektového týmu zruším. Tím se tato signalizace aktivuje, tj. uvedený čas je zapsán jako „Čas aktivace“.
- V čase 28. 3. 2017 v 12:20 systém automaticky odešle mail všem kromě mě a současně se zapíše „Čas uskutečnění“ (signalizace).

5.2.13 Diagram CDM – Šablony

Hlavní (silnou) entitu v tomto diagramu představuje entita „Šablona“, která umožňuje založit události diáře automatizovaně za použití zadaných parametrů, případných typových událostí diáře a případné zadané výchozí události diáře.

Uživatelsky definovatelný číselník, entita „Parametr šablony“, umožňuje uživatelům specifikovat parametry určitého typu (entita „Datový typ“ – viz kapitola „Diagram CDM kategorie diáře“). Parametr kromě zpřesnění rozsahu datového typu nese symbolické jméno, pod kterým bude jeho hodnota vystupovat ve výrazech specifikujících cílové generované události diáře šablonou.

Vazební entita „Parametr přiřazený šabloně“ přiřazuje určité parametry k šabloně. Hodnoty parametrů přiřazené k šabloně bude uživatel zadávat při jejím použití, Tyto hodnoty pak lze použít ve výrazech specifikujících cílové generované události diáře šablonou.



Obrázek 18- Šablony

Každá šablona může mít více pravidel, entita „Pravidlo šablony“. Na základě jednoho pravidla lze vygenerovat jednu cílovou událost diáře, v případě aplikace opakovatelnosti i více souvisejících událostí diáře. Pravidlo šablony obsahuje výrazy, které umožnují specifikovat cílové atributy generované události diáře. V těchto výrazech lze užít běžné funkce použitého programovacího jazyka, symbolická jména parametrů (ta budou při vyhodnocení výrazu nahrazena zadanou hodnotou) a symbolická jména reprezentující atributy připojené typové události diáře nebo zadané zdrojové události.

Je-li atribut „Zadat zdrojovou událost“ nastaven na „Ano“, pak je uživatel při aplikaci šablony vyzván k výběru některé stávající události diáře, aby se z ní cílová událost diáře odvodila. Podobně lze odvodit cílovou událost diáře z typové události diáře (viz entita „Událost diáře“ v tomto diagramu), která je k danému pravidlu připojena. Typová událost diáře má stav „Typová“ (viz entita „Stav události diáře“ v kapitole „Diagram CDM – Událost diáře“).

Programátory definovaný číselník „Měrná jednotka času“ specifikuje, v jakých měrných jednotkách je možné uvádět časovou periodu pro opakovatelnost. Data odpovídající entitě „Měrná jednotka času“ v tabulkách budoucího IS, tedy musí být automaticky vytvořena při instalaci systému. Jedná se zejména o tyto měrné jednotky: Hodina, …, století.

K některým pravidlům může být specifikováno, že se má dle něj vygenerovat více cílových událostí diáře. Toto se specifikuje ve vazební entitě „Specifikace opakovatelnosti“, kde atribut:

- Množství času v MJ – specifikuje velikost periody opakování v dané měrné jednotce času.
- Výraz specifikující termín poslední opakované události – specifikuje do jakého data, tedy za kterým již ne, se budou generovat cílové události diáře.

Pokud je v pravidle atribut „Zapsat vazbu na zdrojovou událost“ nastaven na „Ano“, pak je mezi zadanou zdrojovou událostí diáře a událostmi vygenerovanými díky opakovatelnosti vytvořena vazba typu „Opakování“ (viz entita „Typ vazby objektů“ v kapitole „Diagram CMD – Koncept objektů“).

5.3 Business Process Model

V rámci této práce se BPM zaměřuje na uživatelské postupy při práci s multilingválním diářem, a to zejména s cílem prověřit, zda informační schopnost CDM plně odpovídá požadavkům, resp. z nich vyplývajících předpokládaných uživatelským postupům při práci s IS „Multilingvální diář“.

Při určitém zjednodušení lze říci, že jednotlivé diagramy PDM se zaměřují na pracovní postupy související s instalací systému, se správou uživatelsky definovatelných číselníků, s vlastním užíváním diáře včetně složitějších zápisů pomocí šablon, se signalizací významných událostí diáře a také na automatické úpravy dat (např. přesun nesplněných úkolů na další den). Poměrně velká péče je věnována problematice vyhledání skupinových volných termínů, řešení (potenciálních) časových kolizí v diáři jednotlivých uživatelů a technice upozorňování na významné situace (signalizaci).

V případě správy číselníků se předpokládá standardizované schéma, kdy se zobrazí vybrané záznamy s volbou další akce, například s volbou zobrazení detailních údajů („karty“) aktuálního záznamu s možností jejich aktualizace.

Pro zobrazení diagramů BPM byl v této práci v PD zvolen typ modelu (Model Type) „Business process diagram” s notací (Process Language) „Analysis“, který je intuitivně srozumitelný všem účastníkům včetně zadavatele.

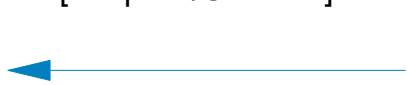
5.3.1 Objekty procesního modelu v PowerDesigner

Jádrem tohoto modelu jsou diagramy, ve kterých jsou elementární procesy (operace, úkony) zobrazeného procesu propojeny vazbami (Flow), které vyjadřujícími směr, jak proces postupuje. Průběh procesu je dále řízen rozhodovacími bloky (možnost větvení) a synchronizací více větví procesu do společného pokračování. V diagramech této práce se používají tyto grafické značky (Symbols) pro jednotlivé druhy objektů:

- Start (začátek) – představuje počáteční událost, která iniciuje (zahajuje, zakládá) proces, je tedy podnětem k zahájení procesu. Může se to být jakákoliv relevantní událost v reálném světě (například zpráva), ale také rozhodnutí uživatele (klik či jiná volba) anebo to, že nastal určitý plánovaný čas. Tato počáteční událost je zdrojovým objektem navazujícího směru průběhu procesu (Flow). V jednom diagramu může být několik různých počátečních událostí, tedy proces může být iniciován několika různými způsoby, případně se může v průběhu již zahájeného procesu čekat (synchronizace) na další vnější událost, které je v diagramu prezentována také tímto symbolem. [12] V této práci nastaveno (Display Preferences), aby se jako součást tohoto symbolu v diagramu zobrazoval jeho název (Name). K úplnému popisu počáteční události je použit komentář.

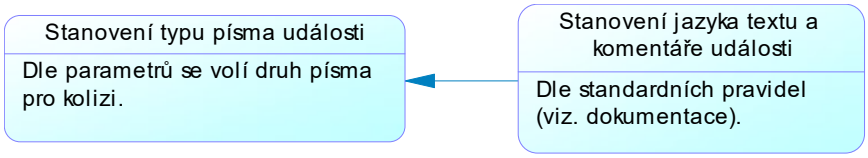
Obrázek 19- Začátek procesu

- Flow (tok – směr pokračování průběhu procesu) – vyjadřuje postup řízení procesu, tj. které procesy (činnosti) a v jakém pořadí budou prováděny. Přehledné zobrazení řízeného průběhu procesu je základním cílem těchto modelů. Tok je symbolizován šipkou, která směřuje od zdrojového objektu k cílovému objektu. Těmito objekty jsou nejčastěji procesy (Process), dále pak rozhodovací bloky (Decisions) a synchronizace (Synchronization). Zdrojovým objektem může být také počáteční událost (Start) a cílovým objektem ukončení procesu (End). V případě, že je zdrojovým objektem rozhodovací blok, pak je součástí zobrazení tohoto symbolu podmínka (Condition - alias), při jejímž splnění proces pokračuje daným směrem ve svém průběhu. [12]

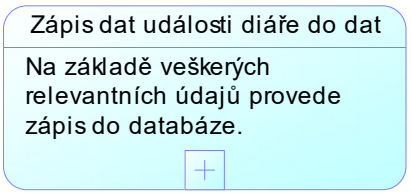


Obrázek 20- Směr pokračování procesu

- Process (proces) – představuje určitou dílčí aktivitu (činnost, sub-proces) v rámci průběhu procesu, který je daným diagramem modelován. Činnost může být prováděna pracovníky (zadání vstupních údajů nové události diáře) anebo zcela automaticky (například odeslání upozorňovacího mailu). V případě, že daný symbol procesu reprezentuje rozsáhlou či složitou činnost, může být tento proces dekomponován (Decomposed Process), tedy modelován samostatným dalším diagramem – pak je součástí symbolu dekomponovaného procesu ikona „+“ a funkce pro zobrazení diagramu. V této práci nastaveno (Display Preferences), aby se jako součást symbolu procesu v diagramu kromě názvu zobrazoval také jeho komentář (Comment), čímž se zvyšuje srozumitelnost a vypovídací schopnost. Do tohoto symbolu a z něj směřují toky (Flow) určující další průběh modelovaného procesu.



Obrázek 22- Ukázka procesů

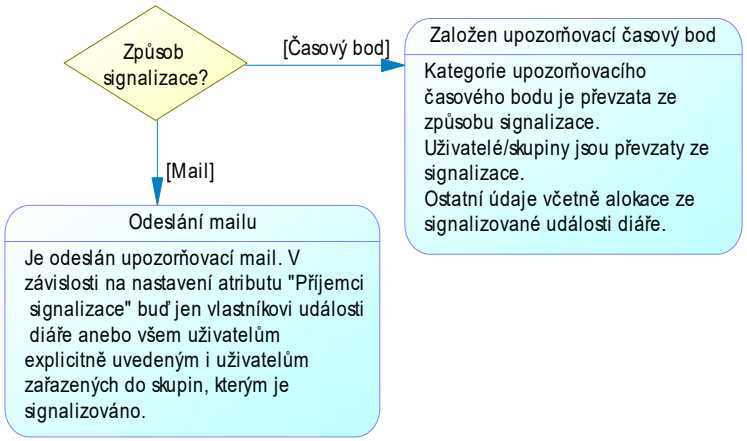


Obrázek 21- Decomposed Process

- Decisions (rozhodovací blok) – umožňuje, aby se na základě zde uvedené podmínky rozhodlo, do které větve bude směřovat pokračování v průběhu procesu. Rozhodovací kritérium (Condition – alias) je zobrazeno jako součást tohoto symbolu (například zda se jedná o „Poslední událost?“). Do rozhodovacího bloku vstupuje jeden nebo více vstupních toků a jeden nebo více výstupních toků vystupuje. Pokud vystupuje jediný tok, pak se nejedná o rozhodnutí, ale jedná se o sloučení vstupních toků, tedy o jednotné pokračování původně různých větví procesu (v takovémto symbolu je jako rozhodovací kritérium uvedeno obvykle jen „OR“). Pokud vystupuje více toků, je každý z nich opatřen podmínkou, při jejímž

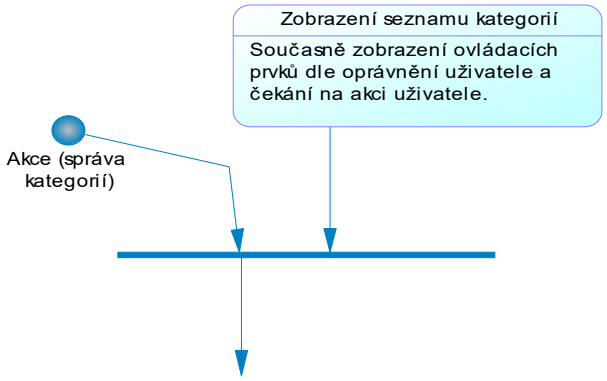
Obrázek 23- Rozhodovací blok

splnění bude proces pokračovat danou větví. Pokud má proces pokračovat všemi výstupními toky, pak se obvykle jako rozhodovací kritérium uvádí jen „AND“. Proces se díky rozhodovacímu bloku může větvit, tedy i pokračovat svým průběhem ve více větvích současně. [12]



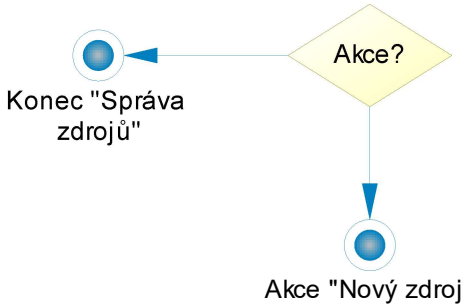
Obrázek 24- Ukázka rozhodnutí

- Synchronization (synchronizace) – zajišťuje synchronizaci více větví procesu, tedy aby se několik vstupních toků současně setkalo v jediném bodě (některé čekají na jiné) a aby se pak společně pokračovalo výstupním tokem (výstupními toky) v provádění procesu. Díky synchronizaci může dojít k pozastavení průběhu procesu anebo ke spojení více větví do společného dalšího průběhu. V této práci se synchronizace obvykle používá v situaci, kdy jsou uživateli prezentovány (zobrazeny) určité údaje a čeká se na jeho reakci. [12]



Obrázek 25- Synchronizace

- End (ukončení procesu) – tento symbol představuje bod, ve kterém daný proces končí svůj průběh. [12] V diagramu jednoho procesu může být několik různých zakončení procesu. V rámci ukončení procesu může vzniknout událost, která může iniciovat start dalšího procesu – pokud je toto modelováno v této práci, pak název ukončení (End) je velmi podobný názvu související události (Start) a tyto názvy jsou uvozeny slovem „Akce“.



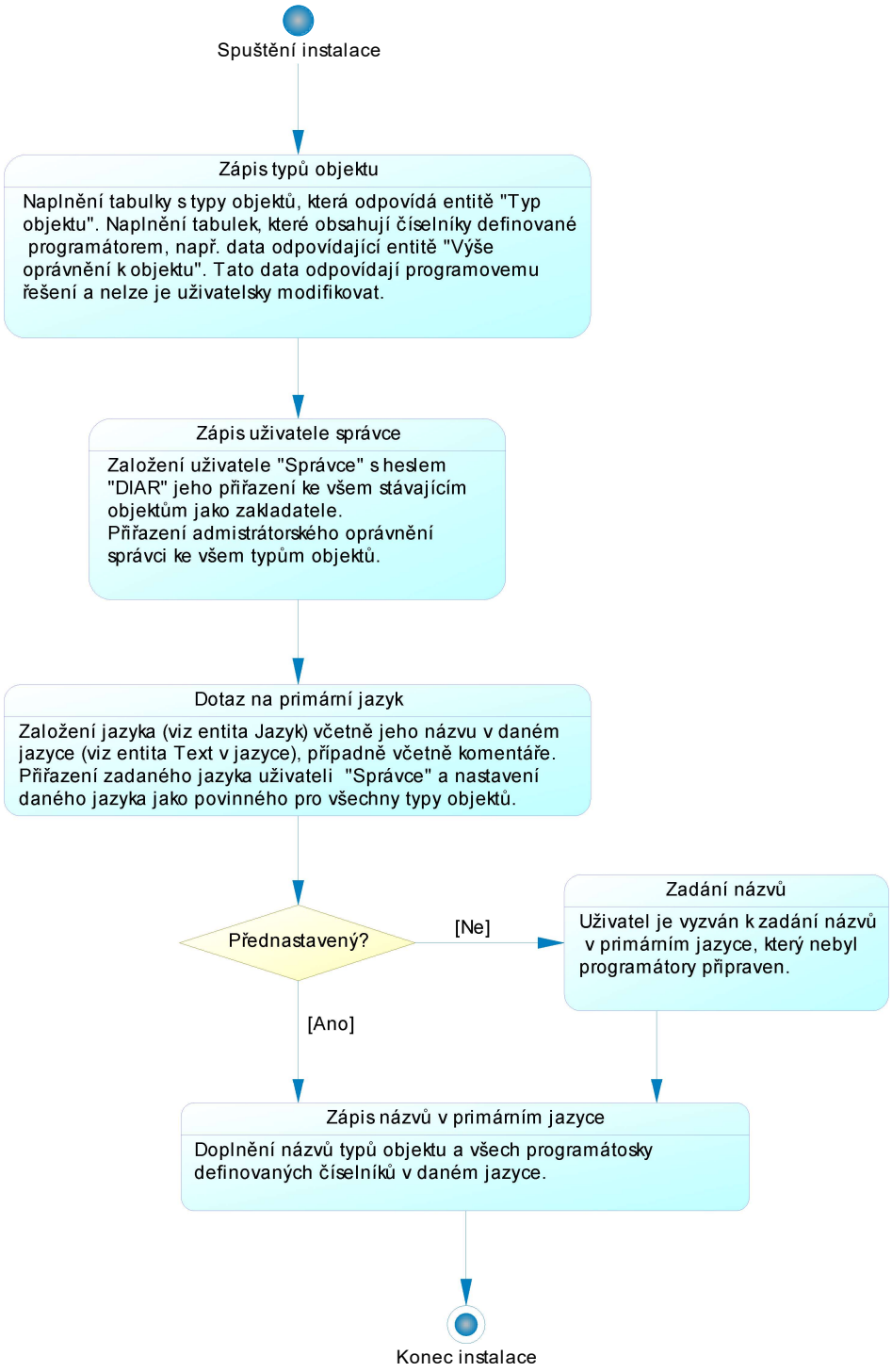
Obrázek 26 - Konec

V BPM a jeho diagramech je možné v této notaci používat ještě další objekty těchto typů:

- Organization unit (organizační jednotka), která iniciuje proces (činnost) anebo je jím oslovována. S tímto procesem je ve vazbě pomocí objektu Role association.
- Resource (zdroj), který může specifikovat data z datových modelů používaná či vytvářená procesem, se kterým je ve vazbě pomocí objektu Resource Flow. [12]

5.3.2 Diagram BPM - Instalace systému

V diagramu Instalace systému jsem se zaměřil převážně na procesní zpracování požadavků, které byly zmíněny v analýze požadavků. Problematika fyzické dislokace databáze a dalších souborů (programy apod.) je ponechána na programovém řešení. Předpokládám, že bude řešena programátory obvyklým způsobem.



Obrázek 27- BPM instalace systému

Instalace, jako počáteční věc využívání systému, klade důraz na zajištění správného fungování systému. Musí tedy být během instalace provedena základní konfigurace systému, což v tomto případě znamená naplnění tabulek databáze daty, které řídí provoz systému.

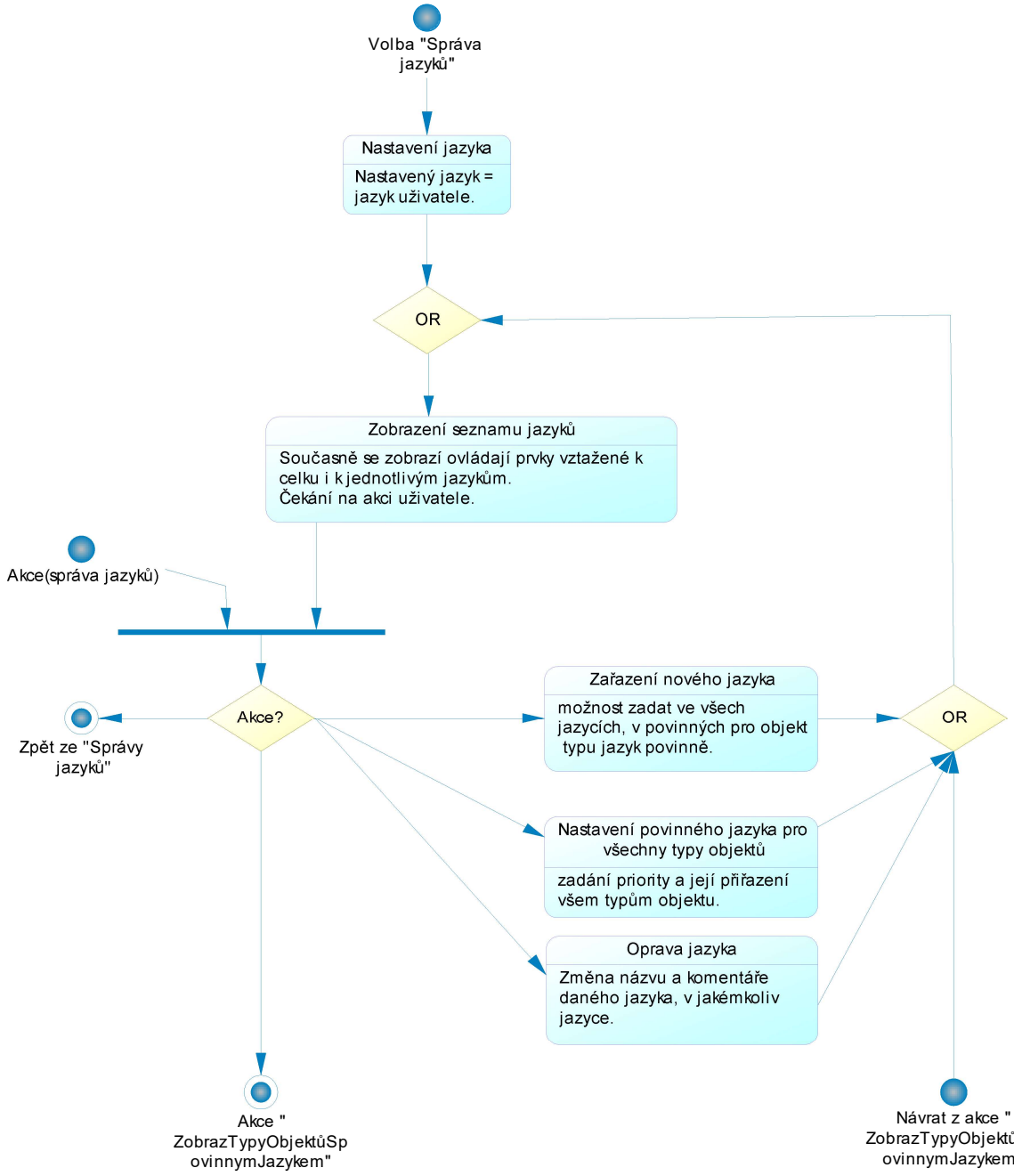
Diagram obsahuje tyto elementární procesy (operace):

- Prvním procesem v diagramu, který následuje po startovací události (Spuštění instalace), je Zápis typů objektů. Tyto záznamy jsou specifikovány programátory (více viz CDM) a fakticky každá entita (tabulka v PDM) představuje jeden typ objektu. Současně jsou naplněny daty tabulky, jejichž obsah jsou číselníky programátory definované.
- Na něj navazuje bezprostředně proces Zápis uživatele správce. Jehož primárním úkolem je založení uživatele "správce" s heslem "DIAR". Jediný tento uživatel se může přihlásit k budoucímu informačnímu systému "multilingvální diář" a nastavit ho pro provoz v dané společnosti. Současně je tento uživatel přiřazen ke všem stávajícím objektům jako jejich autor, čímž je dořešen problém referenční integrity a je možné ji obnovit. Dále je tomuto uživateli nastaveno administrátorské oprávnění ke všem typům objektu.
- Další proces Dotaz na primární jazyk zajistí v komunikaci s uživatelem volbu jazyka, jeho zápis do dat a jeho přiřazení uživateli "správce". Současně je zadaný jazyk nastaven jako povinný pro všechny typy objektu (tedy veškeré texty a komentáře musí být uváděny v tomto jazyce). Jako primární jazyk může být zvolen některý z jazyků, které jsou programátory přednastaveny, anebo jazyk nepřednastavený.
- Pokud je zvolen nepřednastavený jazyk, pak v procesu Zadání názvů je uživatel vyzván, aby zadal názvy (texty) všech typů objektu a všech programátory přednastavených číselníků. Oporou mu je při tom anglické znění těchto textů.
- Následně jsou tyto texty ve zvoleném jazyce zaznamenány do dat, kde budou sloužit uživatelům.

Tímto je instalace dokončena a informační systém je připraven k provozu pro správce. Předpokládá se, že správce bude definovat další uživatele systému a jejich skupiny, zařadí uživatele do skupin, přiřadí jim přístupová oprávnění, bude definovat další případné jazyky a povinnost jejich používání. Fakticky nastaví všechna potřebná data způsobem, který je popsán v dalších diagramech týkajících se správy systému.

5.3.3 Diagram BPM – Správa jazyků

Vzhledem k tomu, že se jedná o multilingvální systém, patří správa jazyků ke klíčovým činnostem správce systému. Specifikace jazyka je popsána v kapitole „Diagram CDM – Multilingvální řešení". Správou jazyků se rozumí zejména: Zařazení nového jazyka, nastavení jazyka (jazyků) jako povinného pro určité typy objektů či pro jednotlivé kategorie diáře.

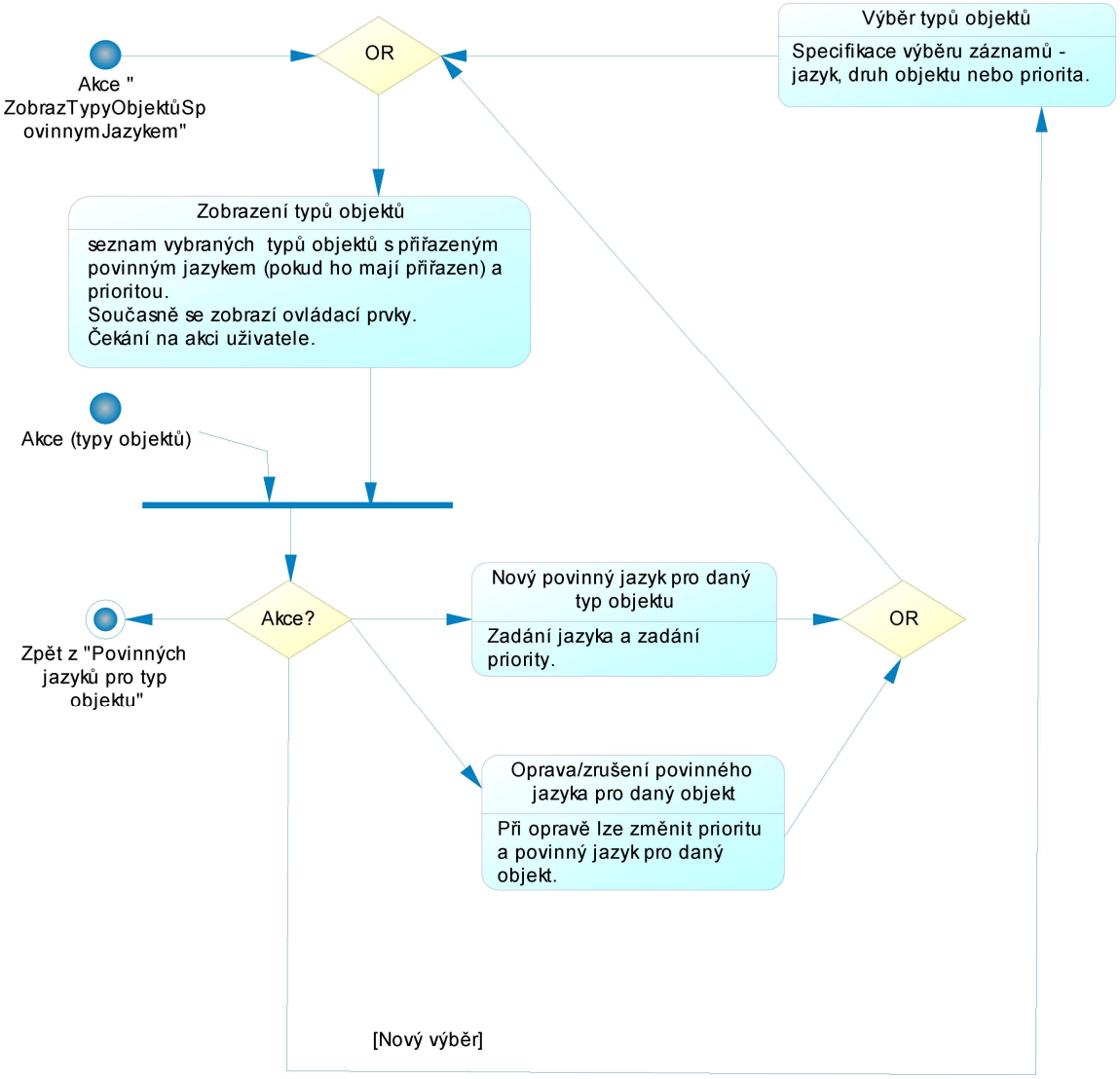


Obrázek 28- BPM správa jazyků

Uživatel může iniciovat správu jazyků z menu volbou relevantní položky menu, přičemž se nastaví jazyk uživatele. Dále se zobrazí seznam stávajících jazyků s ovládacími prvky dle oprávnění uživatele. Uživatel poté může:

- Zařadit nový jazyk a současně musí zadat jeho název a zkratku ve všech povinných jazycích.
- Nastavit daný jazyk jako povinný pro všechny typy objektů včetně priority, v jakém pořadí jazyků budou texty a komentáře zadávány.
- Opravit jazyk, tj. změnit název či komentář vztahující se k danému jazyku.

- Zobrazit seznam typů objektů s povinným jazykem, kde je možné aktualizovat povinnost jazyka pro jednotlivé typy objektů (tedy objekty daného typu musí být opatřeny texty a komentáři v daném jazyce), jak je patrné z následujícího obrázku.



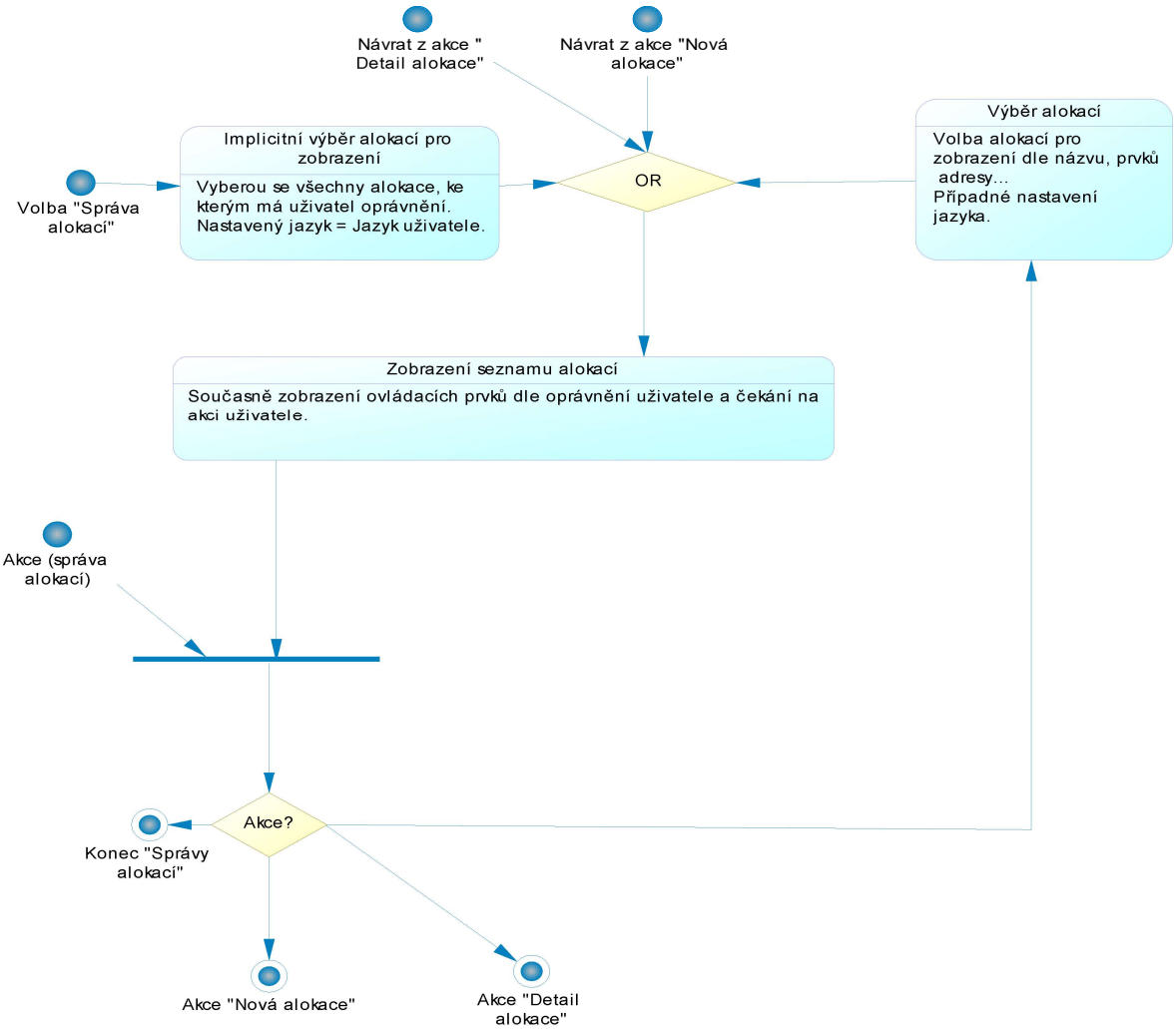
Obrázek 29- BPM zobrazení typů objektů

Obdobně lze zobrazit seznam kategorií s povinným jazykem (viz entita „Povinný jazyk pro událost diáře dané kategorie“), kde je možné aktualizovat povinnost jazyka pro jednotlivé kategorie. Tedy texty a komentáře událostí diáře s danou kategorií musí být uvedeny v daném jazyce.

5.3.4 Diagram BPM – Správa alokací

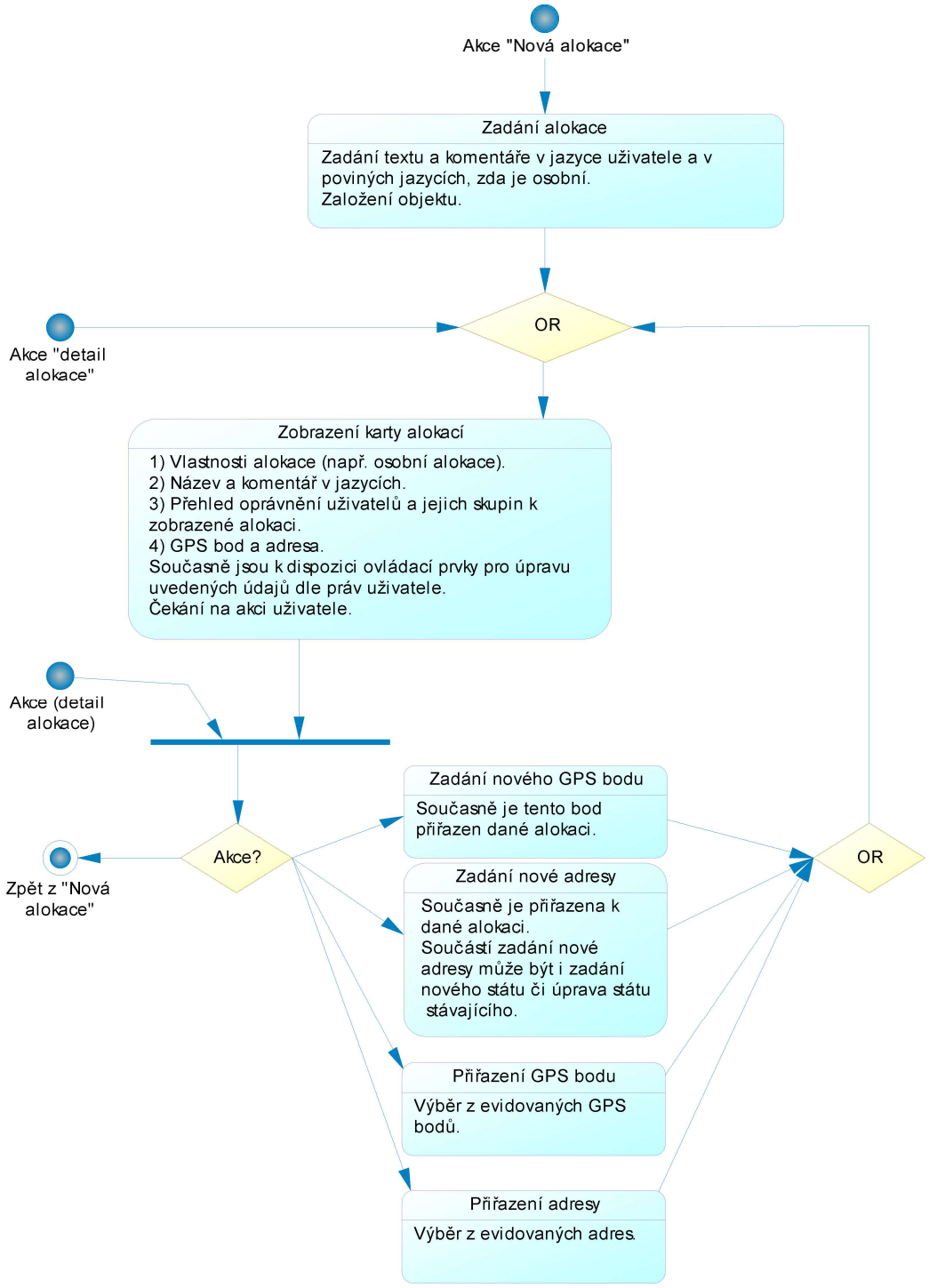
Tento diagram se zaměřuje na specifikaci uživatelských postupů souvisejících se zadáváním nových alokací, úpravou stávajících alokací a na správu souvisejících číselníků. Specifikace alokace je popsána v kapitole „Diagram CDM - Alokace“.

- Uživatel může iniciovat správu alokací z menu volbou relevantní položky menu.
- V tom případě se nastaví, že se mají zobrazit veškeré alokace. Nastaví se jazyk uživatele.
- Dále se zobrazí seznam alokací s ovládacími prvky dle oprávnění uživatele. Uživatel poté může:
    - Změnit původní výběr alokací na jiný seznam alokací například pomocí prvků adresy, nazvu atd. či změnit svůj jazyk.
    - Vybrat alokaci a podívat se na její detail.
    - Založit novou alokaci (viz druhá část tohoto diagramu).
    - Odejít ze správy alokací.



Obrázek 30- BPM správa alokací

Druhá část tohoto diagramu se věnuje jedné konkrétní alokaci, bud´ nové anebo stávající. V případě akce založení nové alokace se nejprve zadají její základní údaje (další údaje jako GPS souřadnice a adresa se přiřazují až následně) a je založena v datech (včetně záznamu v tabulce odpovídající entitě Objekt).



Obrázek 31- BPM detail a nová alokace

Další postup zpracování akce Nová alokace je společný s akcí Detail alokace, tj. zobrazí se karta alokace, která obsahuje zejména tyto údaje:

- Název v nastaveném jazyce.
- Zda je či není osobní a případné další vlastnosti alokace.
- Název a komentář ve všech jazycích, ve kterých jsou uvedeny.
- Přehled oprávnění uživatelů a jejich skupin k zobrazené alokaci.
- Adresa včetně státu a souřadnice GPS.

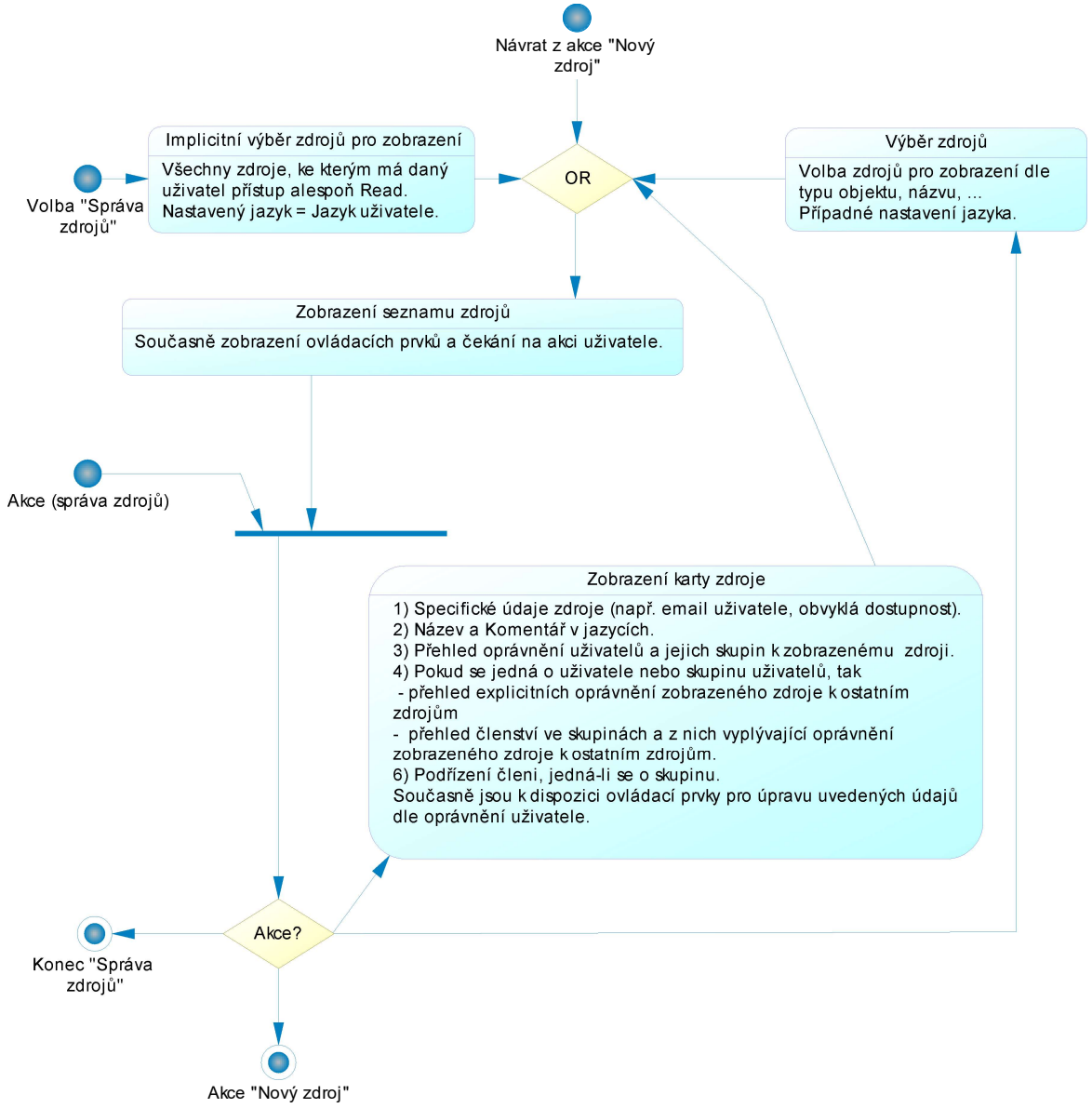
Dále je možné skončit (vrátit se do přehledu alokací, kde již budou patrné provedené změny) anebo dle práv uživatele vybrat akci:

- Zadání nového GPS bodu, který se automaticky ihned přiřadí dané alokaci.
- Přiřazení GPS bodu dané alokaci (přiřadí již evidovaný GPS bod).
- Zadání nové adresy, která se automaticky ihned přiřadí dané alokaci. Současně může být zadán i nový stát.
- Přiřazení adresy k dané alokaci volbou z nabídky již evidovaných adres. Současně může být daná adresa upravena.

V případě výběru jedné z výše uvedených akcí se po jejím ukončení opět zobrazí karta dané alokace včetně nově zadaných či pozměněných údajů. Opět se čeká na akci uživatele.

5.3.5 Diagram BPM - Správa zdrojů

Tento diagram se zaměřuje na specifikaci uživatelských postupů souvisejících se zadáváním nových zdrojů a správou stávajících zdrojů. Specifikace zdroje je popsána v kapitole „Diagram CDM – Zdroje diáře“.



Obrázek 32- Správa zdrojů

Uživatel může iniciovat správu zdrojů z menu volbou relevantní položky menu a zpracování pokračuje těmito procesy:

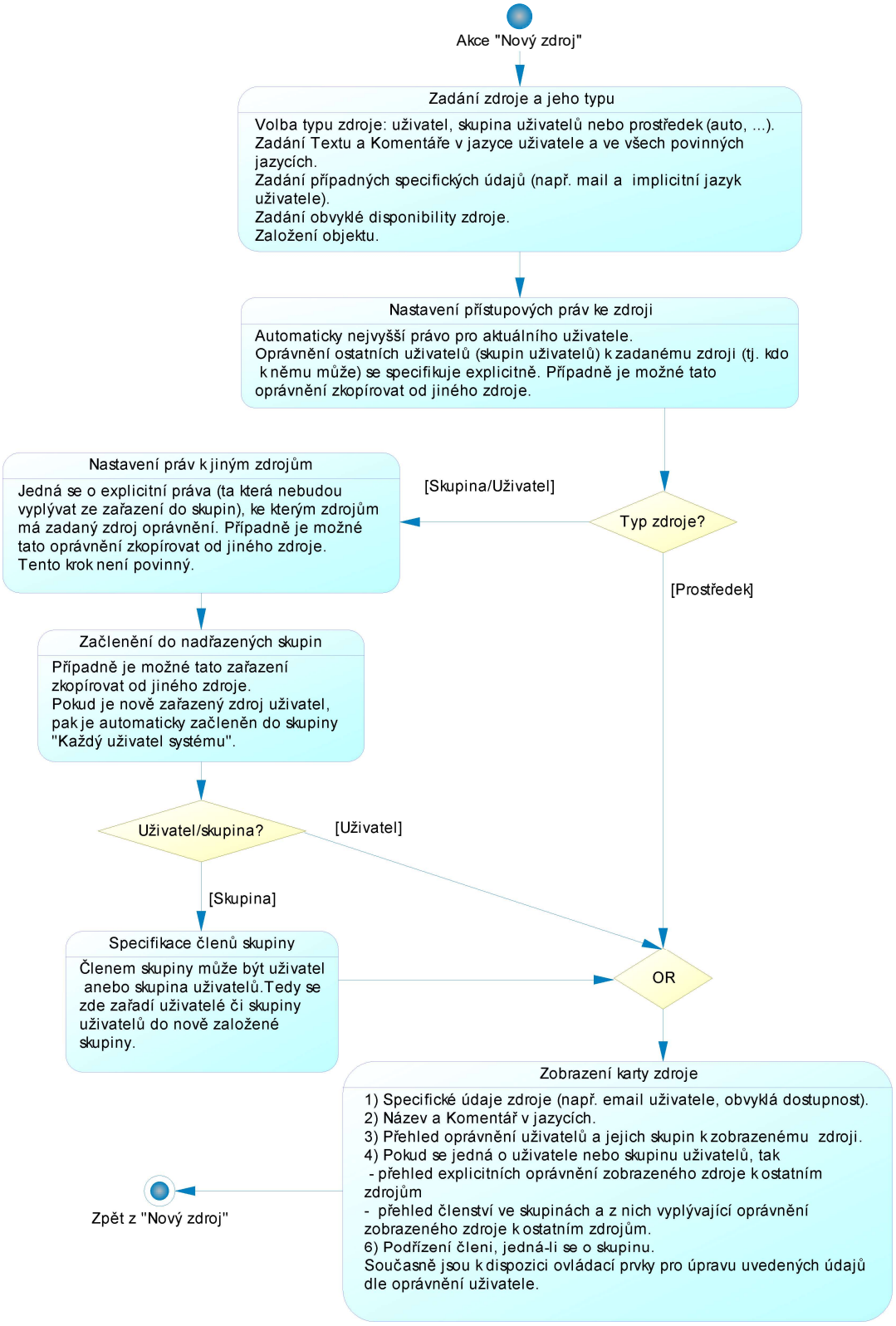
- V tomto případě se nastaví, že se mají zobrazit veškeré zdroje. Dále se nastaví jazyk uživatele pro zobrazení názvu zdrojů, komentářů apod.
- Zobrazí se seznam zdrojů s ovládacími prvky dle oprávnění uživatele a čeká se na aktivitu uživatele. Uživatel poté může:

- Odejít ze správy zdrojů.
- Založit nový zdroj (viz druhá část tohoto diagramu).
- Změnit původní výběr zdrojů na jiný seznam zdrojů například pomocí názvu, typu zdroje (prostředek, uživatel, skupina uživatelů) atd. či nastavit jiný jazyk pro zobrazení údajů.
- Zobrazení karty zdroje, která obsahuje zejména tyto údaje: specifické údaje zdroje, název a komentář v nastaveném jazyce, přehled oprávnění uživatelů a jejich skupin k zobrazení zdroje a ovládací prvky pro úpravu uvedených údajů dle oprávnění uživatele.

Druhá část tohoto diagramu se věnuje akci zadání nového zdroje.

- Prvním procesem v této části diagramu, který následuje po akci nový zdroj, je „Zadání zdroje a jeho typu“. Zadáním typu zdroje se rozumí výběr mezi uživatelem, skupinou uživatelů nebo prostředkem. Dále se zadá Text, Komentář a případné specifické údaje, a to ve všech povinných jazycích.
- Na něj navazuje bezprostředně proces Nastavení přístupových práv ke zdroji. Oprávnění se automaticky nastaví pro aktuálního uživatele na úroveň „Admin“. Oprávnění ostatních uživatelů a jejich skupin k založenému zdroji se specifikují explicitně, případně je možné tato oprávnění zkopírovat od jiného zdroje.
- Pokud založený zdroj představuje uživatele či jejich skupinu, pak se pokračuje procesem Nastavení práv k jiným zdrojům. V tomto procesu se můžou nastavit práva ke zdrojům, která nebudou vyplývat ze zařazení do skupin.
- V navazujícím procesu může být založený zdroj (skupina/uživatel) zařazen do nadřazených skupin. Je-li zařazený zdroj uživatel, je automaticky zařazen do skupiny „Každý uživatel systému“, což mu přiřadí implicitní nastavení parametrů systému.
- Pokud založený zdroj představuje skupinu uživatelů, pak se pokračuje procesem Specifikace členů skupiny. Do skupiny se mohou přiřadit uživatelé či skupiny uživatelů.

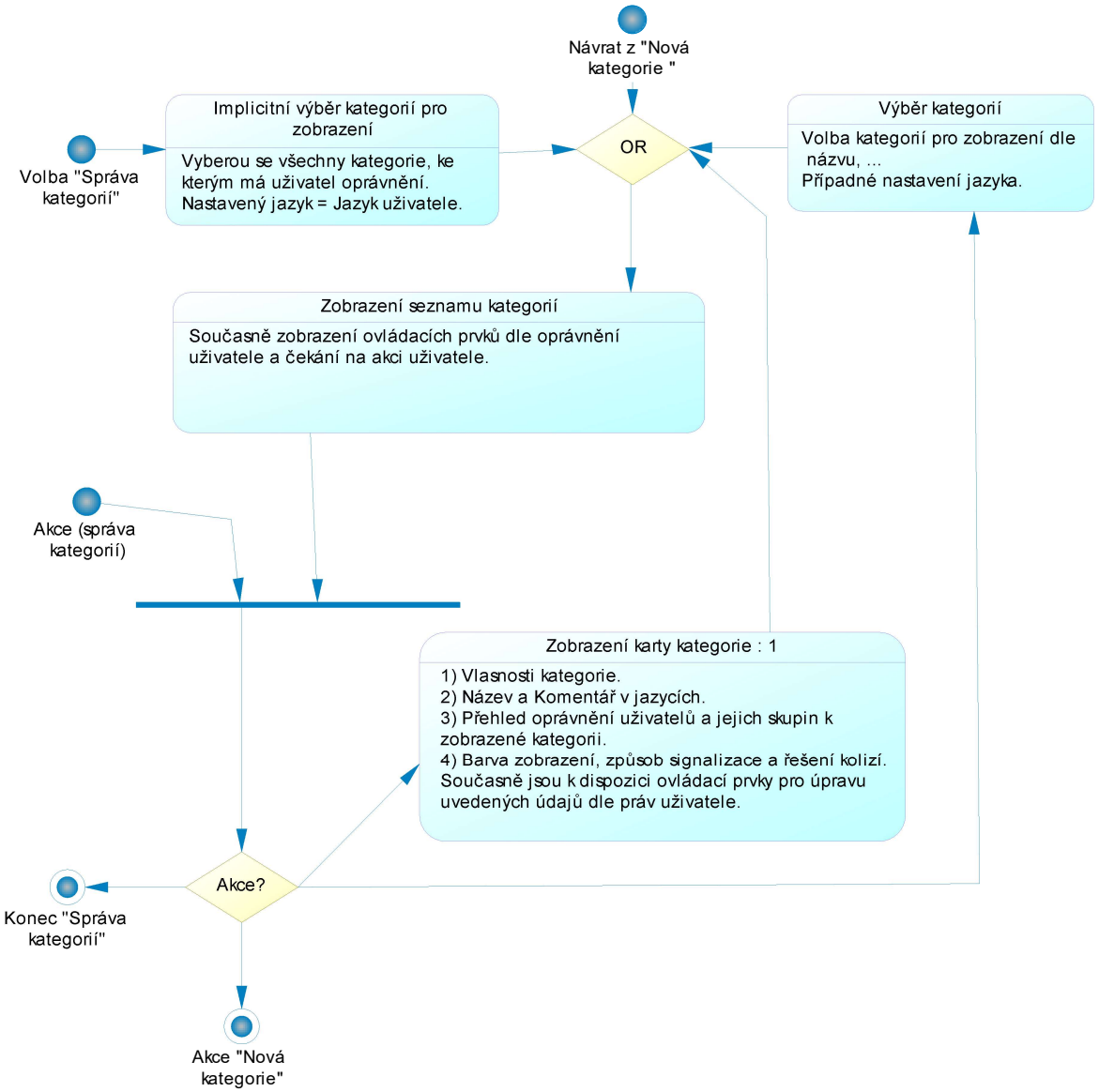
Po zařazení jakéhokoliv nového zdroje se aktivuje Zobrazený karty zdroje (popsáno výše). Poté je možné skončit (vrátit se do seznamu zdrojů, kde již budou patrné provedené změny).



Obrázek 33- Nový zdroj

5.3.6 Diagram BPM – Správa kategorií

Tento diagram se zaměřuje na specifikaci uživatelských postupů souvisejících se zadáváním nových kategorií a správou stávajících kategorií a na správu souvisejících číselníků. Je potřebné si uvědomit, že vlastnosti kategorie mají významný dopad na relevantní události diáře. Specifikace kategorie je popsána v kapitole „Diagram CDM – Kategorie diáře“.



Obrázek 34- Správa kategorií

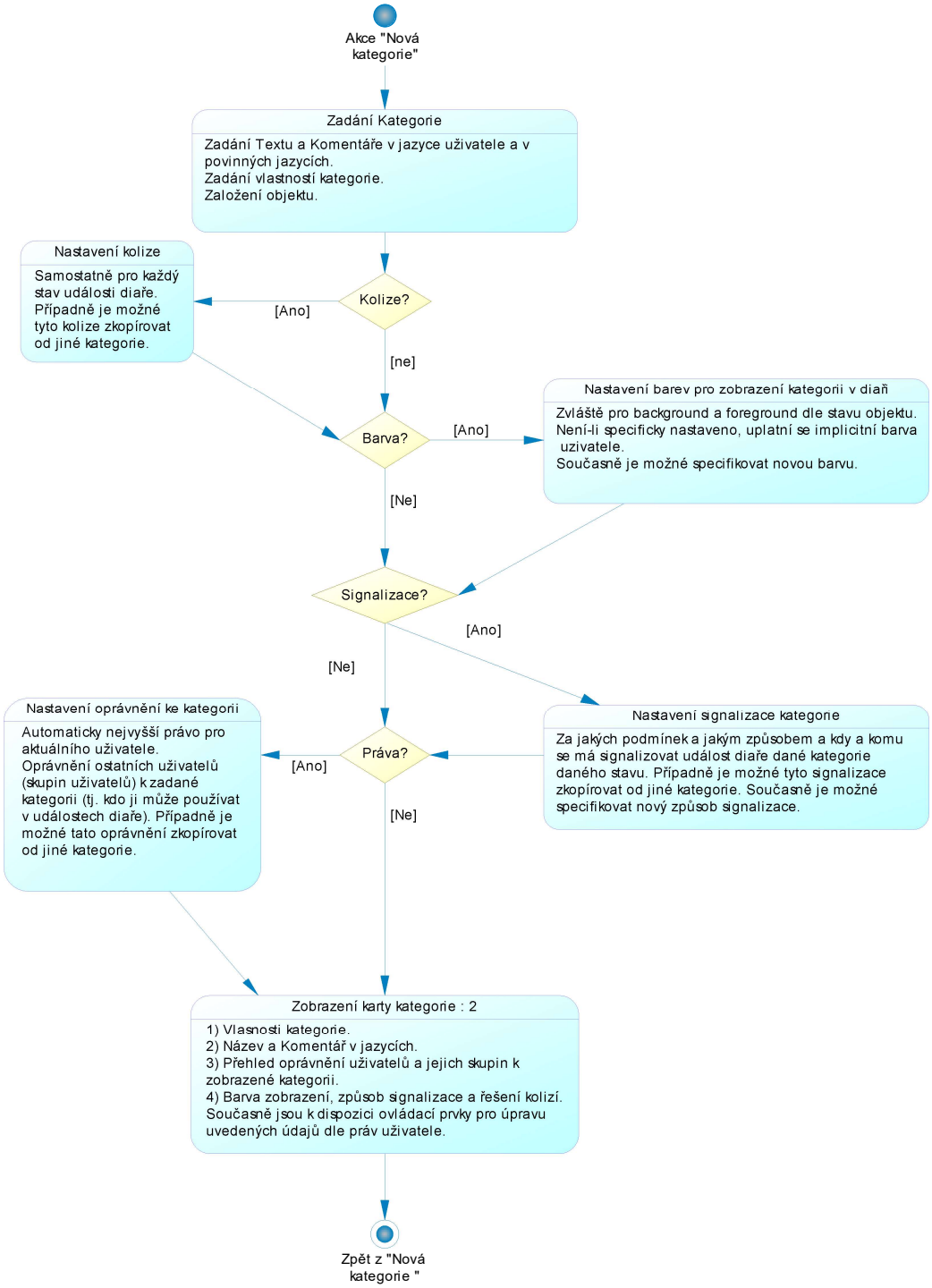
Uživatel může iniciovat správu kategorii z menu volbou relevantní položky menu a zpracování pokračuje těmito procesy:

- V tomto případě se nastaví, že se mají zobrazit veškeré kategorie, ke kterým má daný uživatel oprávnění. Dále se nastaví jazyk uživatele pro zobrazení názvu kategorií, komentářů apod.

- Zobrazí se seznam kategorií s ovládacími prvky dle oprávnění uživatele a čeká se na aktivitu uživatele. Uživatel poté může:
    - Odejít ze správy kategorií.
    - Založit novou kategorii (viz druhá část tohoto diagramu).
    - Změnit původní výběr kategorií na jiný výběr kategorií, které se mají zobrazit, například pomocí názvu, současně lze nastavit jiný jazyk pro zobrazení údajů.
    - Zobrazení karty kategorie, která obsahuje zejména tyto údaje: Vlastnosti kategorie, název a komentář v nastaveném jazyce, přehled oprávnění uživatelů a jejich skupin ke kategorii, barva zobrazení události diáře dané kategorie, způsob signalizace a řešení kolizí dané kategorie, současně jsou k dispozici ovládací prvky pro úpravu zobrazených údajů dle oprávnění uživatele.

Druhá část tohoto diagramu se věnuje akci zadání nová kategorie.

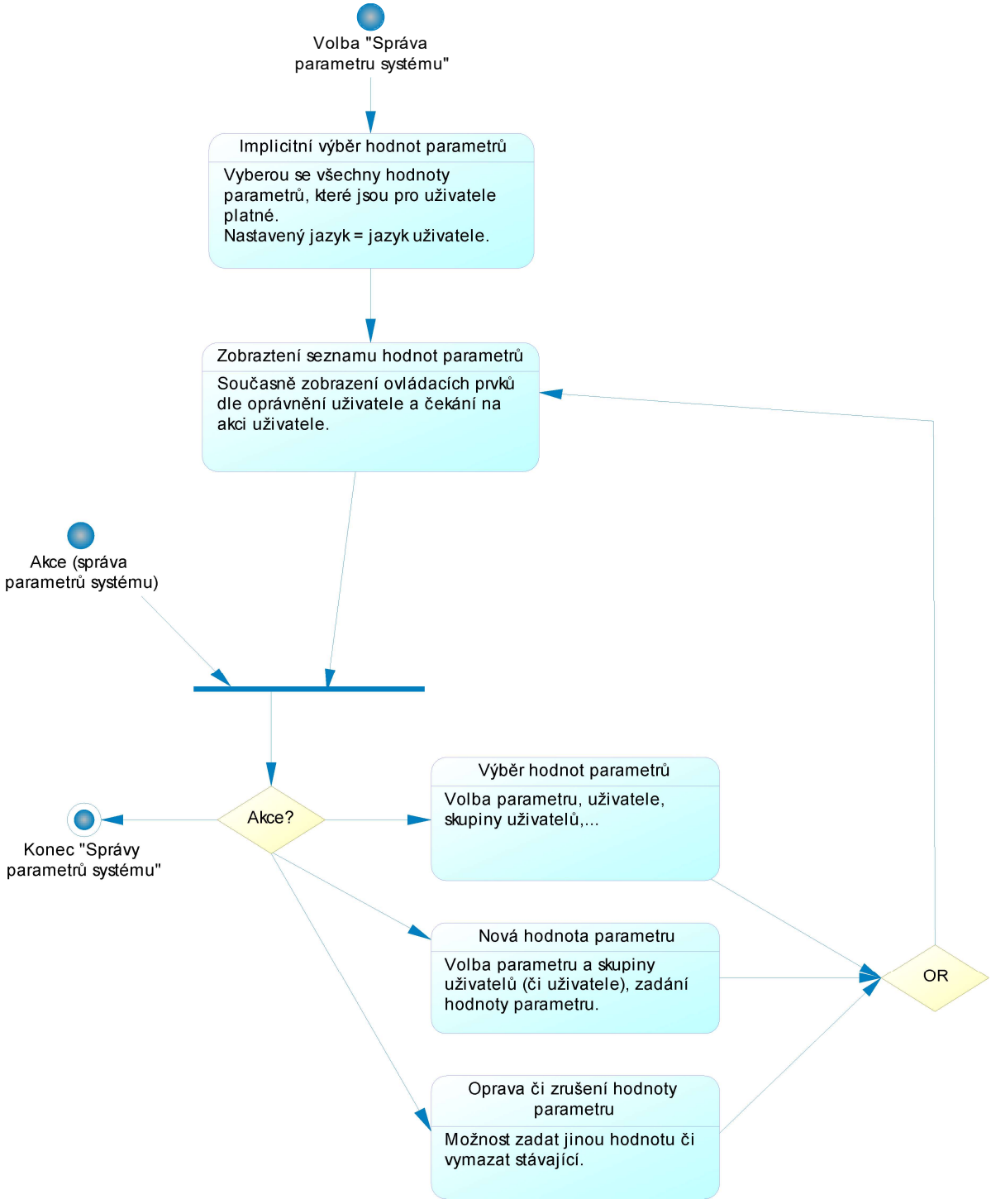
- Prvním procesem v této části diagramu je „Zadání kategorie“. Zadá se Text a Komentář (ve všech povinných jazycích), současně se zadají specifické údaje. Případně se specifikují jazyky, které se musí povinně požít pro popis událostí diáře dané kategorie.
- Pokud u nově založené kategorie chceme nastavit způsob řešení kolize, je definován. Případně je možné způsob řešení kolizi zkopírovat od jiné kategorie.
- Pokud u dané kategorie nechceme implicitní barvu (písma a výplně), je možné nastavit jinou barvu (viz entita „Barva“ v CDM diagramu – Kategorie diáře) pro zobrazení v diáři pro každého uživatele individuálně či skupinu uživatelů, případně specifikovat novou barvu.
- Pokud chceme u dané kategorie nastavit signalizace v diáři, tak se nastaví, za jakých podmínek a jakým způsobem se májí realizovat. Též je možné specifikovat nové způsoby signalizace. Případně je možné specifikace signalizací zkopírovat od jiné kategorie.
- Pro aktuálního uživatele se vždy automaticky nastaví administrátorská oprávnění k dané kategorii. Dále je možné nastavit oprávnění ostatních uživatelů či skupin k zadané kategorii. Případně je možné tato oprávnění zkopírovat od jiné kategorie.
- Po nastavení práv k nové kategorii se aktivuje Zobrazený karty kategorie (popsáno výše). Poté je možné skončit (vrátit se do seznamu kategorií, kde již budou patrné provedené změny).



Obrázek 35- Nová kategorie

5.3.7 Diagram BPM – Správa parametrů systému (diáře)

Tento diagram se zaměřuje na specifikaci uživatelských postupů souvisejících se specifikací hodnot parametrů systému pro jednotlivé uživatel či jejich skupiny. Specifikace parametrů a jejich hodnot je popsána v kapitole „Diagram CDM – Parametry diáře“.



Obrázek 36- Správa parametru systému

Uživatel může iniciovat správu parametrů systému z menu volbou relevantní položky menu a zpracování pokračuje těmito procesy:

- V tomto případě se nastaví, že se mají zobrazit hodnoty parametrů, které se uplatňují pro aktuálního uživatele. Dále se nastaví jazyk uživatele pro zobrazení názvu parametrů, komentářů apod.
- Zobrazí se seznam hodnot parametrů systému s ovládacími prvky dle oprávnění uživatele a čeká se na aktivitu uživatele. Uživatel poté může:
    - Odejít ze správy parametrů systému.
    - Přiřadit novou hodnotu parametru systému uživateli či skupině uživatelů.
    - Opravit (zrušit) stávající hodnotu parametru systému přiřazenou uživateli či skupině uživatelů.
    - Změnit původní výběr hodnot parametrů systému na jiný, například pomocí názvu parametru systému, uživatele, skupiny uživatelů atd., či nastavit jiný jazyk pro zobrazení údajů.

5.3.8 Diagram BPM – Zobrazení diáře

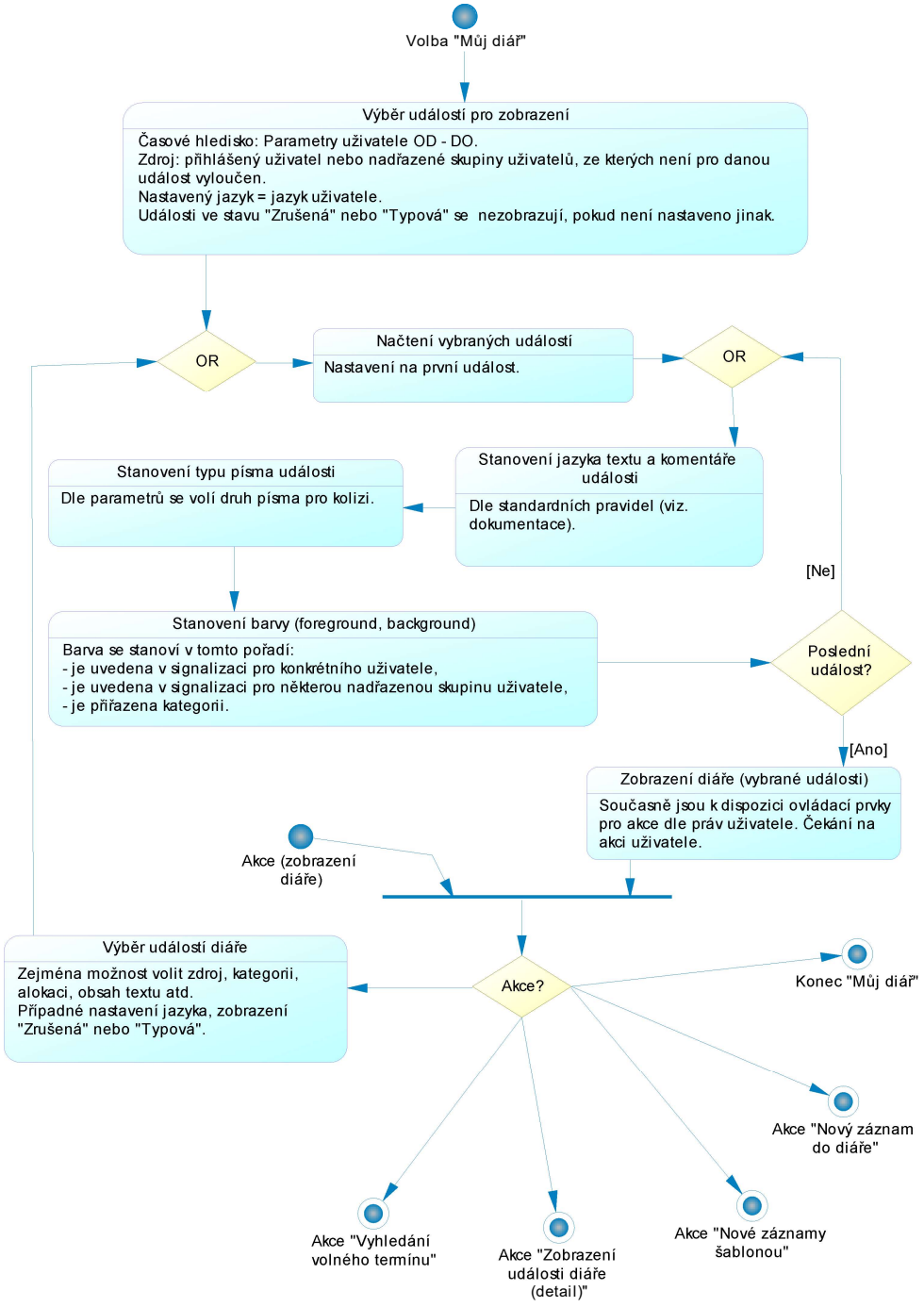
Tento diagram popisuje zobrazení diáře (diářů) uživateli s možností dalších akcí zejména pro aktualizaci událostí diáře, což je zřejmě nejdůležitější aktivita celého navrhovaného IS.

Primárně se zobrazuje diář aktuálního uživatele, případně si zvolí diář jiného zdroje. Pravidla a možnosti zobrazení diáře jsou popsána v kapitolách „Diagram CDM – Kategorie diáře“ a „Diagram CDM – Události diáře“.

Uživatel může iniciovat zobrazení svého diáře volbou relevantní položky menu a zpracování pokračuje těmito procesy:

- V tomto případě se automaticky nastaví období od-do podle parametrů systému, ze kterého se zobrazí události diáře aktuálního uživatele. Současně se nastaví jazyk uživatele pro zobrazení textů, komentářů apod.
- Poté se načtou jednotlivé vybrané události diáře, u kterých se postupně nastaví:
    - Jazyk textu a komentáře, ve kterém bude událost zobrazena (viz pravidla v kapitole „Analýza požadavků -> Multilingvální systém“).
    - Způsob písma se nastaví dle stavu dané události diáře, případně a to prioritně dle signalizace kolize. Viz kapitola „Diagram CDM – Parametry diáře“.
    - Dále se stanoví barva výplně a pozadí. Pravidla pro stanovení barev jsou uvedena v kapitole „Diagram CDM – Kategorie diáře“.

- Vybrané události diáře se zobrazí v daném jazyce, barvě a způsobu písma. Očekává se, že si uživatel zvolí pro další zpracování jednu z následných akcí:
    - Nový záznam do diáře viz diagram Nová událost diáře.
    - Detailní zobrazení události diáře, viz diagram Zobrazení události diáře (detail).
    - Vyhledání volného termínu, viz diagram Vyhledání volného termínu.
    - Nové záznamy šablonou viz diagram Aplikace šablon.
    - Ukončit zobrazení diáře.



Obrázek 37- Zobrazení diáře

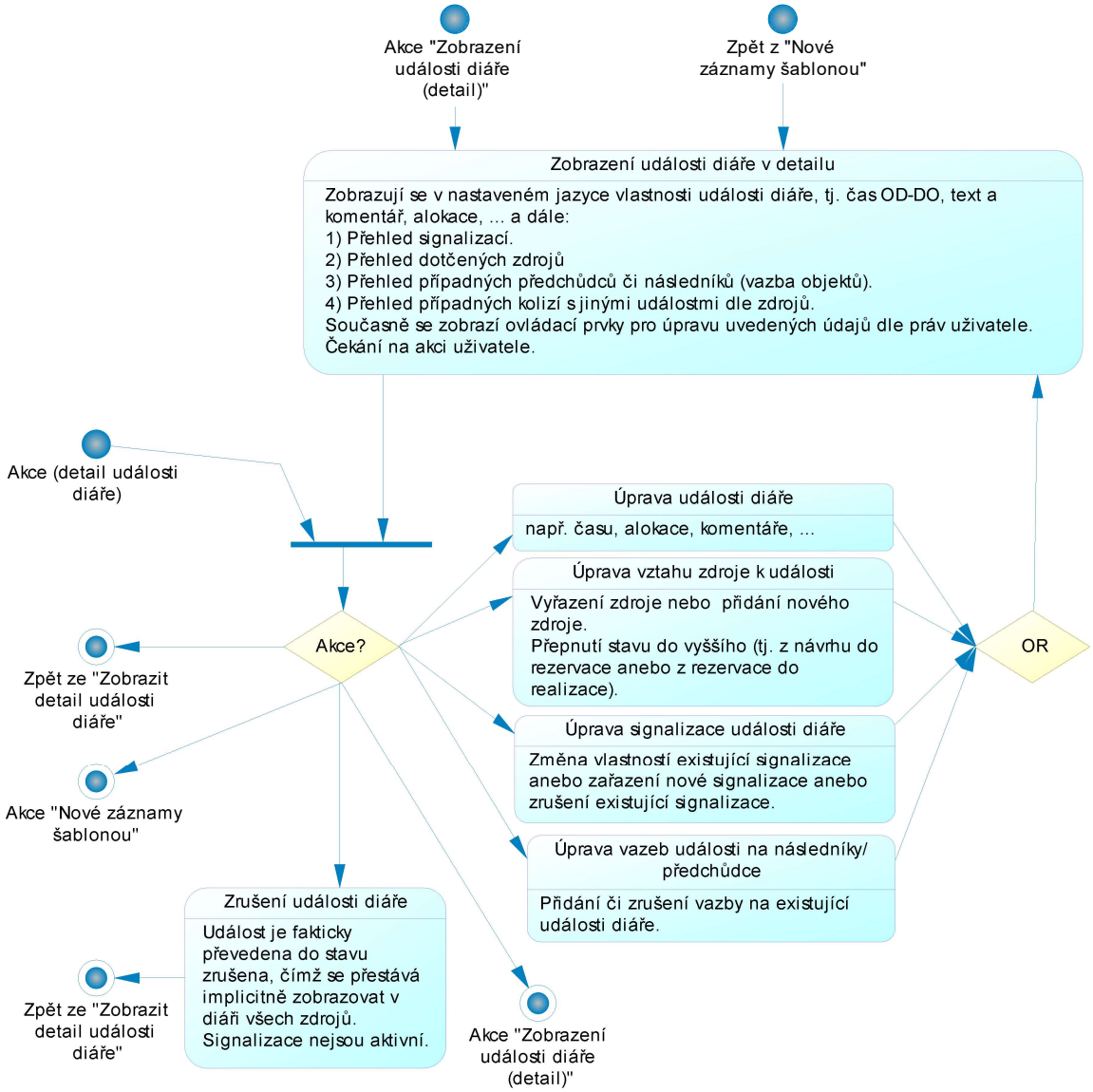
5.3.9 Diagram BPM – Zobrazení události diáře (detail)

Tento diagram se zaměřuje na detail zobrazení události diáře uživateli či skupině uživatelů a jeho úpravu. Zobrazení detailu můžeme iniciovat ze seznamu zobrazených událostí nebo z aplikace šablon.

Při otevření detailu události diáře se zobrazí text a komentář v nastaveném jazyce události diáře. Dále se nám zobrazí k dané události přehled: signalizací, dotčených zdrojů, případných předchůdců či následníků a kolizí s jiným událostí dle zdrojů.

Uživatel poté dle svých práv bude moci:

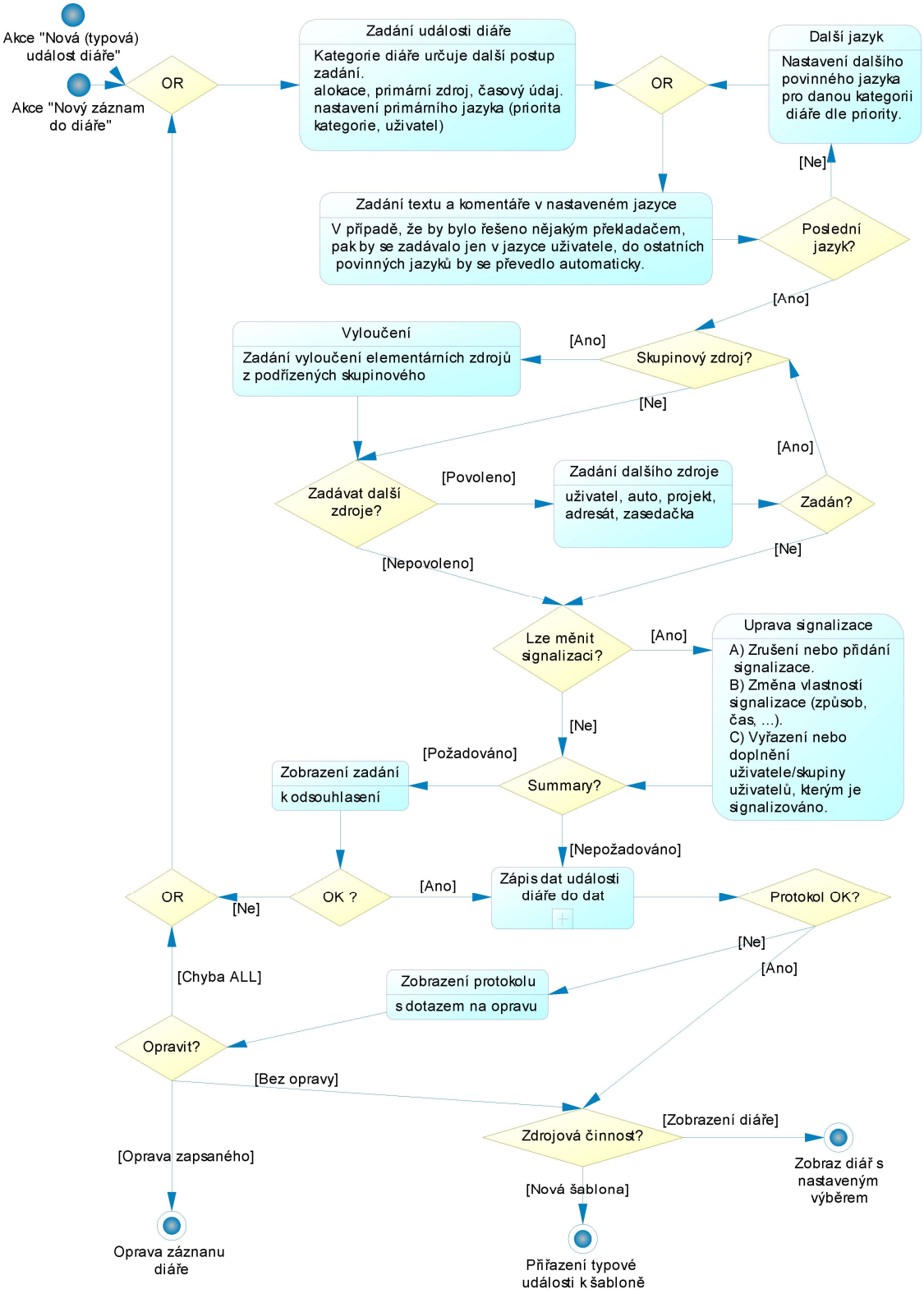
- Upravit událost diáře například změnit komentář.
- Upravit vztah zdroje k události např. přiřazení nového zdroje anebo přepnout z návrhu do rezervace.
- Upravit signalizaci události, změnou vlastností dané události anebo přiřazení nové signalizace anebo zrušení existující signalizace.
- Upravit vazbu události na následníky a předchůdce, tj. přidání či zrušení vazby na existující události diáře.



Obrázek 38- BPM zobrazeni diáře (detail)

5.3.10 Diagram BPM – Nová událost diáře

Tento diagram se zaměřuje na specifikaci postupu při zadání nové události diáře. Tento postup nemusí být zcela triviální díky přiřazení více zdrojů k události, díky textům a komentářům ve více jazycích, díky signalizacím (rozsah a způsob) i díky řadě parametrů, které toto zadání ovlivňují.



Obrázek 39- BPM nová událost diáře

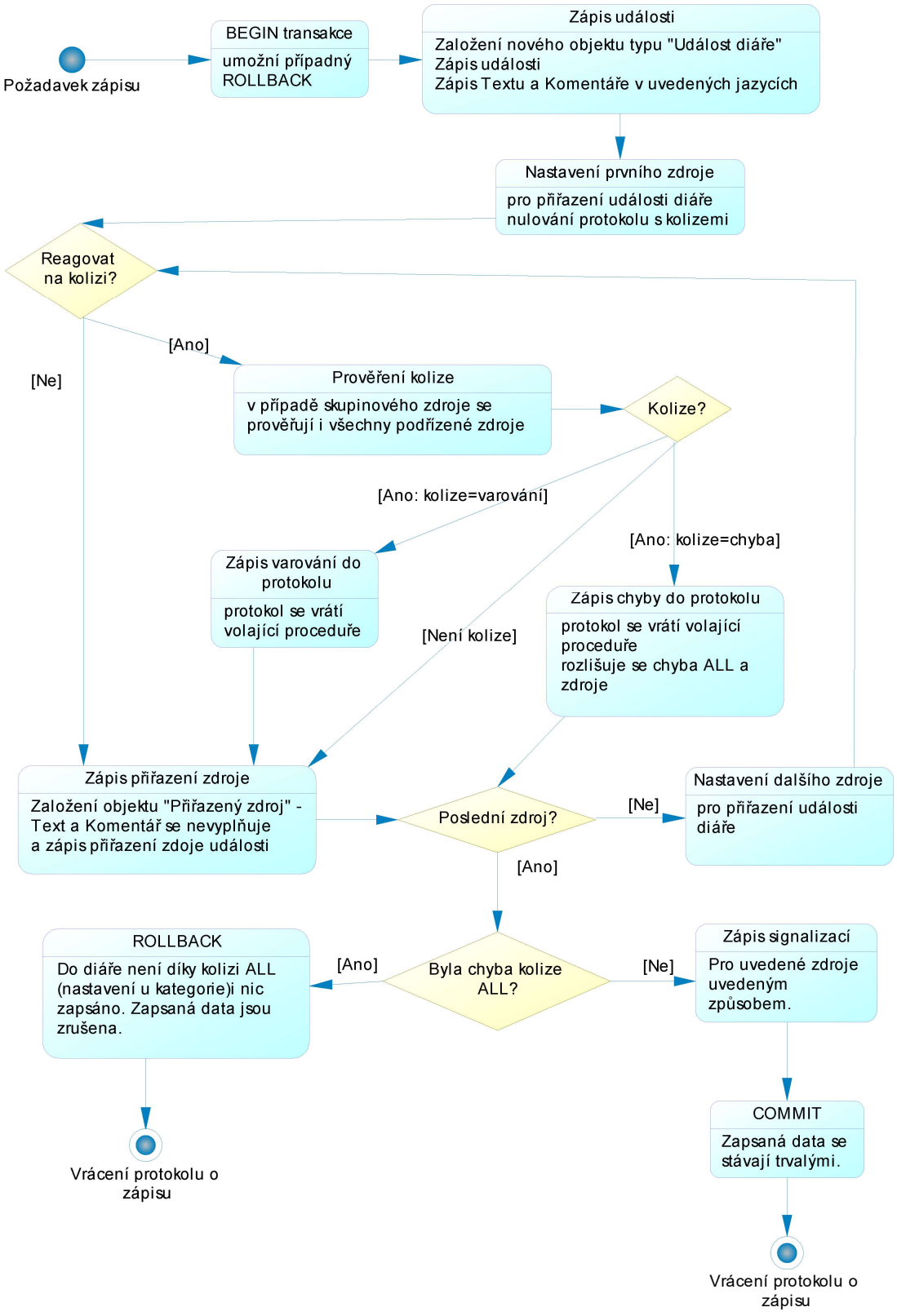
Uživatel může iniciovat založení nové události z menu volbou relevantní položky anebo ze správy šablon (detail šablony) volbou založení nové typové události a dále se pokračuje těmito procesy:

- Zadání elementárních údajů nové události diáře (například kategorie, čas zahájení a ukončení, první zdroj, alokace, …).
- Zadání textu a komentáře v nastaveném jazyce (viz pravidla v kapitole „Analýza požadavků -> Multilingvální systém“).
- Pokud se nejedná o poslední povinný jazyk pro události dané kategorie, pak uživatel zadá popis události (text a komentář) v dalším povinném jazyce.
- Pokud je zadaný zdroj skupina uživatelů a současně je pro danou kategorii povoleno vyloučit uživatele ze skupinového zdroje (viz atribut „Umožnit vyloučení ze skupinového zdroje“ entity „Kategorie diáře“), tak je uživateli umožněno některého člena této skupiny vyloučit.
- Pokud je povoleno zadávat další zdroje (viz atribut „Lze zadávat další zdroje“ entity „Kategorie diáře“), je uživateli umožněno přiřadit k zadávané události další zdroj, a to i opakovaně.
- Pokud je povoleno měnit signalizaci (viz atribut „Lze měnit signalizaci“ entity „Kategorie diáře“), je uživateli umožněno změnit signalizaci, která by se implicitně nastavila pro událost diáře dané kategorie.
- Pokud je požadováno zobrazení sumáře (viz atribut „Zobrazovat sumář před zápisem“ entity „Kategorie diáře“), zobrazí se komplexní přehled, jaké údaje budou zapsány (událost diáře, zdroje kterých se týká, signalizace). Na jejich základě se uživatel může rozhodnout, že je opraví, tedy se vrátí na počátek zadání údajů události diáře.
- Uskuteční se vlastní zápis do dat s řadou kontrol, zejména kolizí. Tento zápis je popsán vlastním diagramem, proto je v tomto diagramu zobrazen s ikonou „+“ značící, že se jedná rozložitelný proces (Decomposed process). Je popsán níže.
- Pokud jsou v protokolu o zápisu kolize, pak se tento protokol zobrazí a proces pokračuje některou z těchto větví:
    - Jedná-li se o kolizi „Při kolizi nezapsat nic“, pak se zápis neuskutečnil a uživatel může zadat upravené údaje znovu.
    - Jedná-li se o jiný způsob řešení kolize, pak se uživatel rozhodne, zda údaje opraví anebo ponechá.
- Nakonec se tato činnost ukončí a uživatel je vrácen do zdrojové činnosti (odkud bylo zadání nové události diáře iniciováno).

Proces „Zápis dat události diáře do dat“, který se v tomto diagramu prezentuje jedním „okénkem“, má svůj vlastní diagram[1]. Na rozdíl od ostatních diagramů vyjadřující uživatelské postupy tento diagram je technicky zaměřen, postup tedy odpovídá zejména dodržení referenční integrity mezi

[1] Tento diagram je dostupný v menu daného objektu (procesu) pod položkou „Open diagram“.

jednotlivými tabulkami budoucího IS. Je iniciován požadavkem na komplexní zápis údajů související s nově založenou událostí diáře, následuje:



Obrázek 40- BPM zápis

- Zahájení transakce, tedy se buď všechna data zapíší anebo ne.
- Zapíše se nový objekt, nová událost diáře, její popisy (text a komentář) ve všech jazycích.
- Pro jednotlivé zdroje, kterých se událost týká:
    - Pokud se způsob řešení kolize „Typ reakce na kolizi“ liší od „Nekontrolovat kolize“, pak se prověří, zda dochází v diáři daného zdroje ke kolizi.
    - Pokud dochází ke kolizi, pak se v závislosti na způsobu řešení kolize zapíše do protokolu buď varování („Typ reakce na kolizi“ má hodnotu „Při kolizi zapsat a upozornit“) anebo chyba.
    - V případě, že není vyhodnocena kolize jako chyba, je událost přiřazena k danému zdroji.
- Po zpracování posledního zdroje se rozhodne o pokračování:
    - V případě, že nastala kolize, jejímž řešením je „Při kolizi nezapsat nic“, jsou zrušena zapsaná data (technicky zřejmě ROLLBACK).
    - V opačném případě jsou zapsány signalizace dané události, data se stávají trvalými (technicky zřejmě COMMIT).
- Nakonec je zdrojovému procesu vrácen protokol o zápisu události diáře.

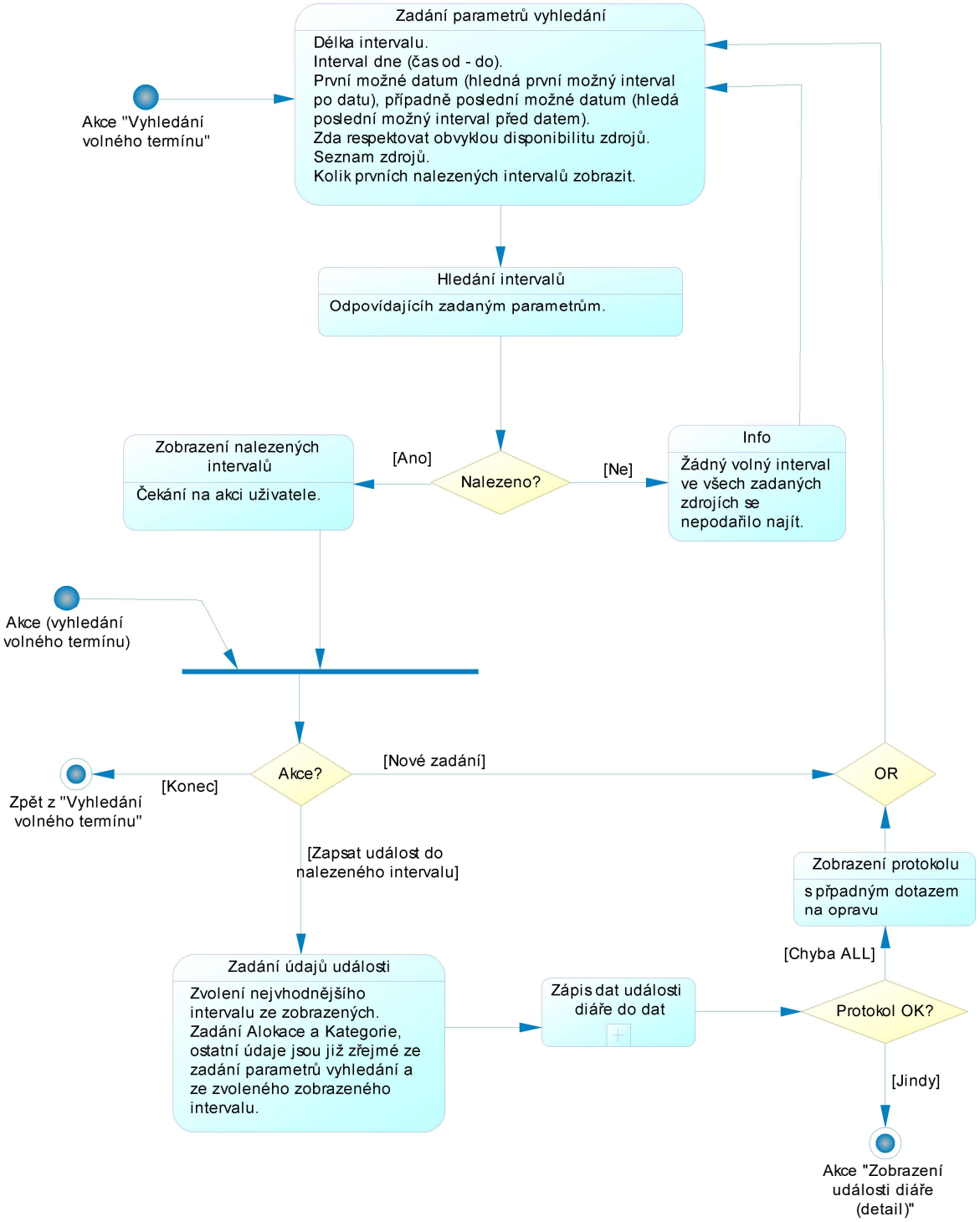
5.3.11 Diagram BPM – Vyhledání volného termínu

Tento diagram se zaměřuje na specifikaci postupu při vyhledávání volného termínu pro událost v diáři. Uplatní se zejména pro plánování události pro více uživatelů, aby nedošlo k případné kolizi.

Uživatel může iniciovat vyhledání volného termínu z menu volbou relevantní položky a dále se pokračuje těmito procesy:

- Zadání parametrů vyhledání volného termínu:
    - Délka intervalu – souvislý časový interval vyjádřený počtem hodin a minut.
    - Čas od a do – denní čas (např. 12:00 – 16:00), ve kterém se má hledat volný prostor.
    - První možné datum – je-li uvedeno, pak se hledá volný termín po tomto. Typické pro metodu určení času začátku „as soon as possible“.
    - Poslední možné datum – je-li uvedeno, pak se hledá první volný termín před tímto datem. Typické pro metodu určení času konce „as late as possible“. Nehledá se do minulosti za aktuální čas. Nelze kombinovat s prvním možným datem.
    - Zda respektovat obvyklou disponibilitu zdrojů – nalezený volný termín musí být v čase, kdy všechny relevantní zdroje jsou obvykle disponibilní, viz entita „Obvyklá disponibilita zdroje“.
    - Kolik intervalů zobrazit – je-li nalezen zde uvedený počet volných intervalů, je hledání ukončeno.
    - Seznam zdrojů – jedná se o zdroje, které musí mít současně ve svém diáři volný termín.
- Uskuteční se vyhledání volných termínů dle zadaných parametrů.

- Pokud se nenajde žádný volný interval, pak se uživateli zobrazí zpráva o neúspěchu a bude moci vyhledat jiný termín zadáním jiných parametrů vyhledání.
- V případě úspěšného vyhledání volných intervalů, se zobrazí seznam nalezených intervalů s ovládacími prvky a čeká se na aktivitu uživatele. Uživatel poté volí:
    - Odejít z vyhledání volného termínu.
    - Zadat nové parametry vyhledání volného termínu.
    - Zapsat událost do nalezeného intervalu.
- Při volbě zápisu události do zvoleného nejvhodnějšího intervalu ze zobrazených se zadá alokace a kategorie, ostatní údaje se převezmou ze zadaných parametrů.
- Uskuteční se vlastní zápis do dat s řadou kontrol, zejména kolizí. Tento zápis je popsán vlastním diagramem, proto je v tomto diagramu zobrazen s ikonou „+“ značící, že se jedná rozložitelný proces (Decomposed process). Je popsán v „Diagramu BPM – Nová událost diáře“.
- Pokud jsou v protokolu o zápisu události diáře vážné chyby (kolize se způsobem řešení „Při kolizi nezapsat nic“ – zřejmě někdo jiný mezitím obsadil nalezený termín), pak se tento protokol zobrazí. Po jeho prohlédnutí může uživatel zadat nové parametry vyhledání.
- Pokud při zápisu nedošlo k vážné kolizi, tak se tato činnost ukončí a uživateli se zobrazí nová událost diáře (detail). Zde je možné učinit případné další úpravy nové události diáře (nová signalizace, vyloučení uživatele, …).



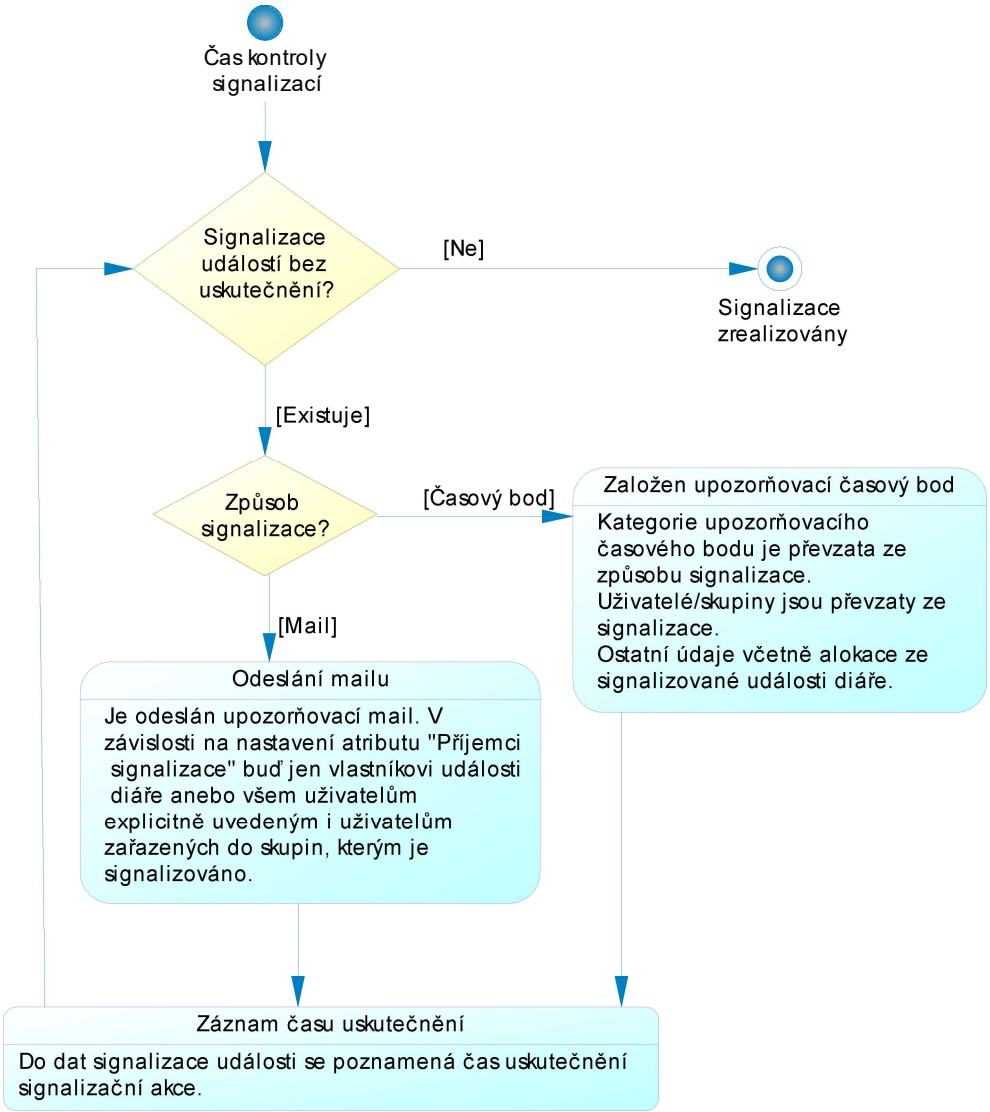
Obrázek 41- BPM vyhledání nového termínu

5.3.12 Diagram BPM – Signalizace

Tento diagram se zaměřuje na specifikaci postupu, kterým je automaticky (tj. ze strany IS a nikoliv ze strany uživatele) realizována signalizace události diáře způsobem email a upozorňovací časový bod, a postupu, kterým se automaticky posouvá na aktuální datum nesplněný úkol, aby nezapadl v minulosti.

První část diagramu je zahájena časem kontroly signalizací, který specifikuje, kdy se má tento proces/postup aktivovat (např. se může jednat o event databázového systému s periodou 2 minut). Diagram obsahuje tyto elementární procesy (operace):

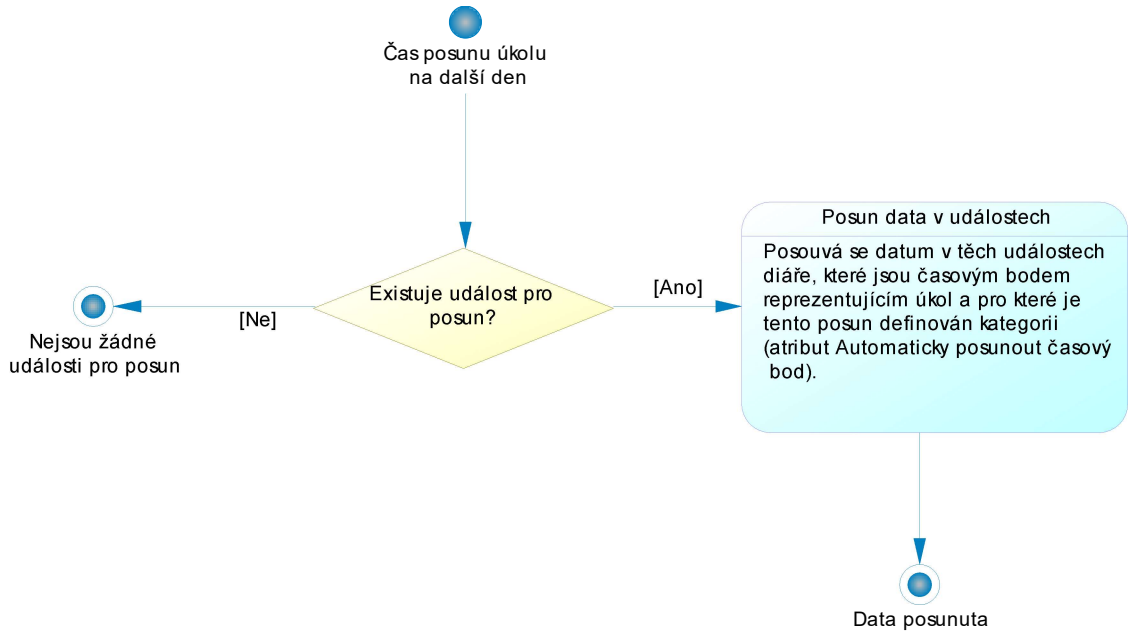
- V případě existence signalizace bez záznamu času uskutečnění, ale s uvedeným časem aktivace, se realizuje signalizace dle způsobu jejího provedení:
    - Založen upozorňovací časový bod.
    - Odeslán upozorňovací email.
- Poté se zaznamená čas uskutečnění (viz entita „Signalizace události“). A signalizace se zrealizuje.
- Toto je prováděno v cyklu, dokud existují aktivované, ale neuskutečněné, signalizace.



Obrázek 42- BPM Signalizace

Druhá část diagramu se věnuje „nezapadnutí nesplněného úkolu“. V čase posunu úkolu na další den (viz diagram CDM – Parametry diáře, např. ve 23:59) se zahájí tento proces/postup (opět lze předpokládat databázový event, který se aktivuje každý den v uvedeném čase):

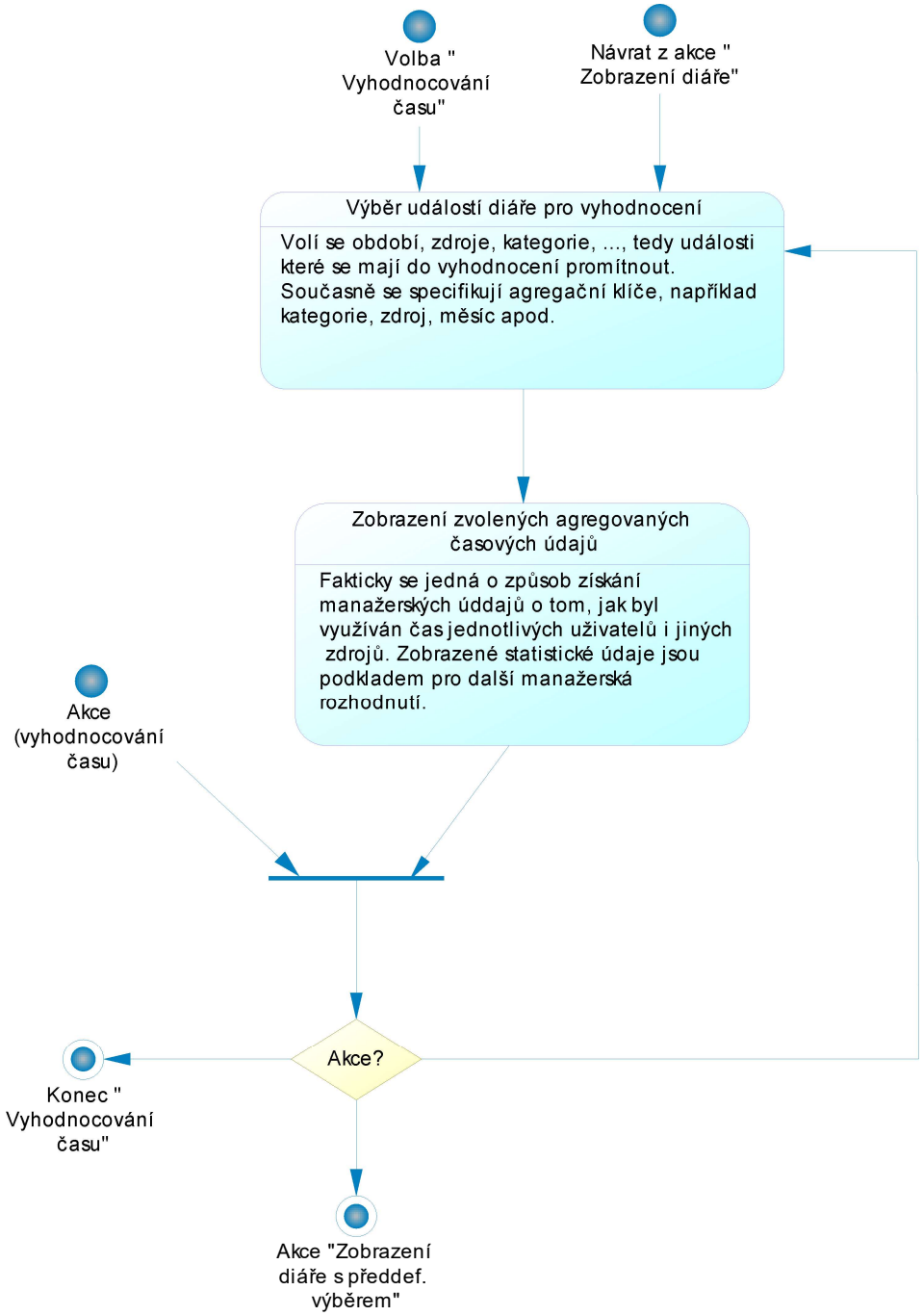
- V případě existence událostí diáře typu časový bod (obvykle úkol, tj. kategorie s atributem „Automaticky posunout časový bod“ nastaveným na „Ano“), které jsou ve stavu plánovaná či navržená a jejichž čas zahájení je nižší než aktuální čas, posune se jejich datum vpřed.
- V případě neexistence uvedených událostí se neprovede nic.



Obrázek 43- BPM posun úkolu

5.3.13 Diagram BPM – Vyhodnocování času

Tento diagram popisuje vyhodnocování času (Time management) uživatelů či jiných zdrojů. Je zřejmé, že v diáři jsou evidovány údaje o množství času plánovaném i uskutečněném dle jednotlivých kategorií, přičemž kategorie mohou představovat projekty, procesy či další sledované aktivity, a dle jednotlivých zdrojů, přičemž tyto zdroje mohou být uspořádány do skupin (organizačních celků).



Obrázek 44- BPM vyhodnocování času

Analýzou agregovaných údajů o využití času lze získat řadu užitečných informací, které lze použít pro další manažerská rozhodování. Fakticky tento proces tedy představuje manažerský systém zaměřený na vyhodnocení spotřeby času s případnou analýzou efektivnosti. Uživatel iniciuje proces vyhodnocování času z menu volbou relevantní položky.

- V tomto případě uživatel nastaví (vybere), jakou množinu události bude chtít vyhodnotit (dle data, kategorií, zdrojů, alokace, stavu událostí apod.) a podle jakých kritérií se mají elementární údaje agregovat.
- Zobrazí se seznam se zvolenými agregovanými údaji (vyhodnocení), například dle uživatelů, dalších zdrojů, kategorií apod. Očekává se některá z těchto akcí od uživatele:
    - Zobrazit události diáře s přednastaveným výběrem na ty, které přispěly do agregovaného údaje. Následuje návrat do této činnosti.
    - Výběr jiné množiny analyzovaných dat či jiných agregačních kritérií.
    - Ukončení této činnosti.

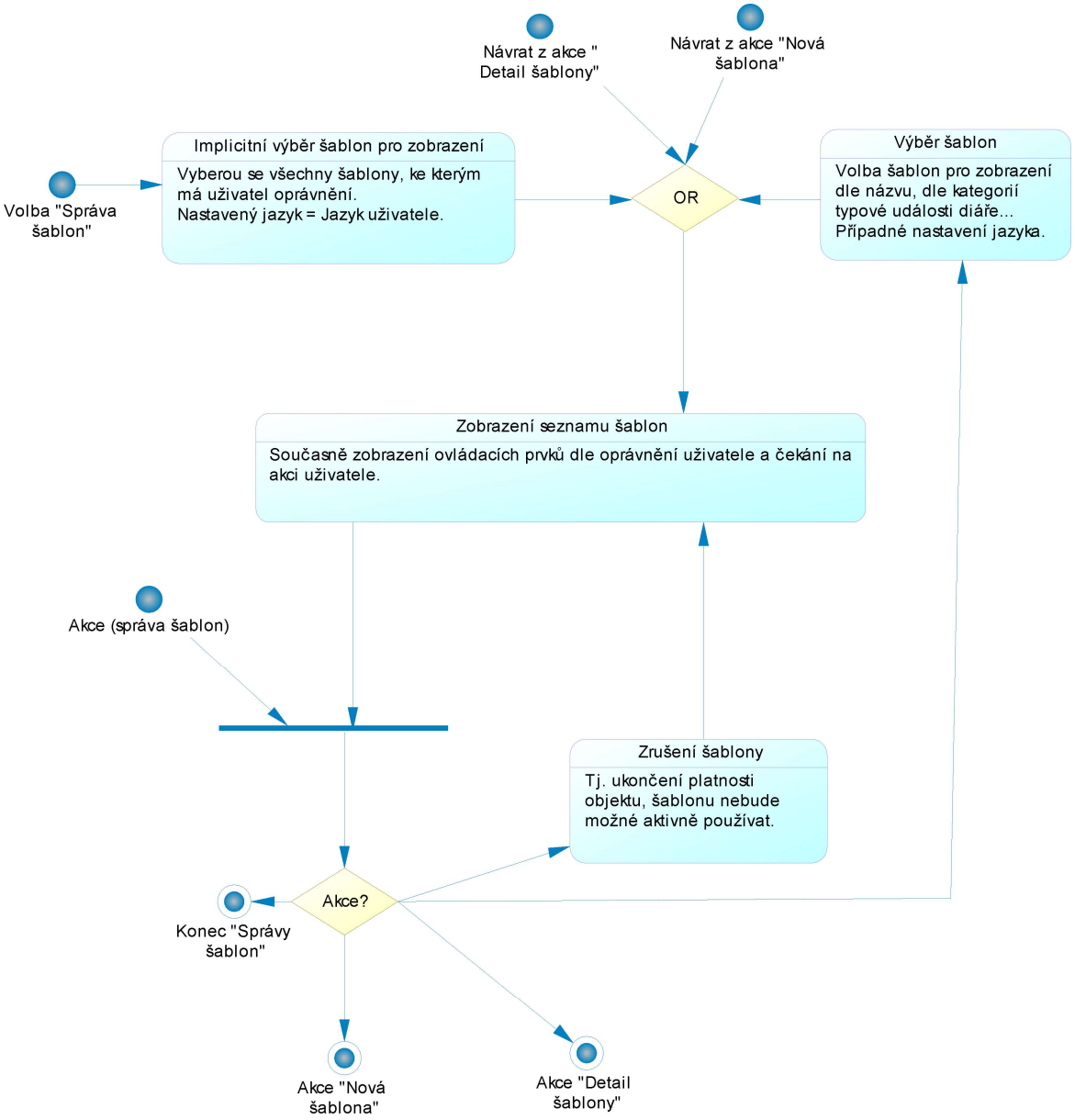
5.3.14 Diagram BPM – Správa šablon

Tento diagram se zaměřuje na specifikaci uživatelských postupů souvisejících se zadáváním nových šablon a správou stávajících šablon. Specifikace šablon a jejich vlastností je popsána v kapitole „Diagram CDM – Šablony“. Správa šablon lze rozdělit na dvě aktivity:

- Přehled stávajících šablon s případnou aktivací druhé aktivity.
- Zobrazení detailních údajů jedné šablony, případně založení nové šablony.

Uživatel může iniciovat správu šablon z menu volbou relevantní položky menu a zpracování pokračuje těmito procesy:

- V tomto případě se automaticky nastaví, že se mají zobrazit veškeré šablony, ke kterým má uživatel oprávnění. Dále se nastaví jazyk uživatele pro zobrazení názvu šablon.
- Zobrazí se seznam šablon s ovládacími prvky dle oprávnění uživatele a čeká se na aktivitu uživatele. Uživatel poté může:
    - Odejít ze správy šablon.
    - Založit novou šablonu (viz druhá část tohoto diagramu).
    - Změnit původní výběr šablon na jiný seznam šablon, například pomocí názvu, či nastavit jiný jazyk pro zobrazení údajů.
    - Zrušit šablonu.
    - Vybrat šablonu a podívat se na její detail (viz druhá část tohoto diagramu).



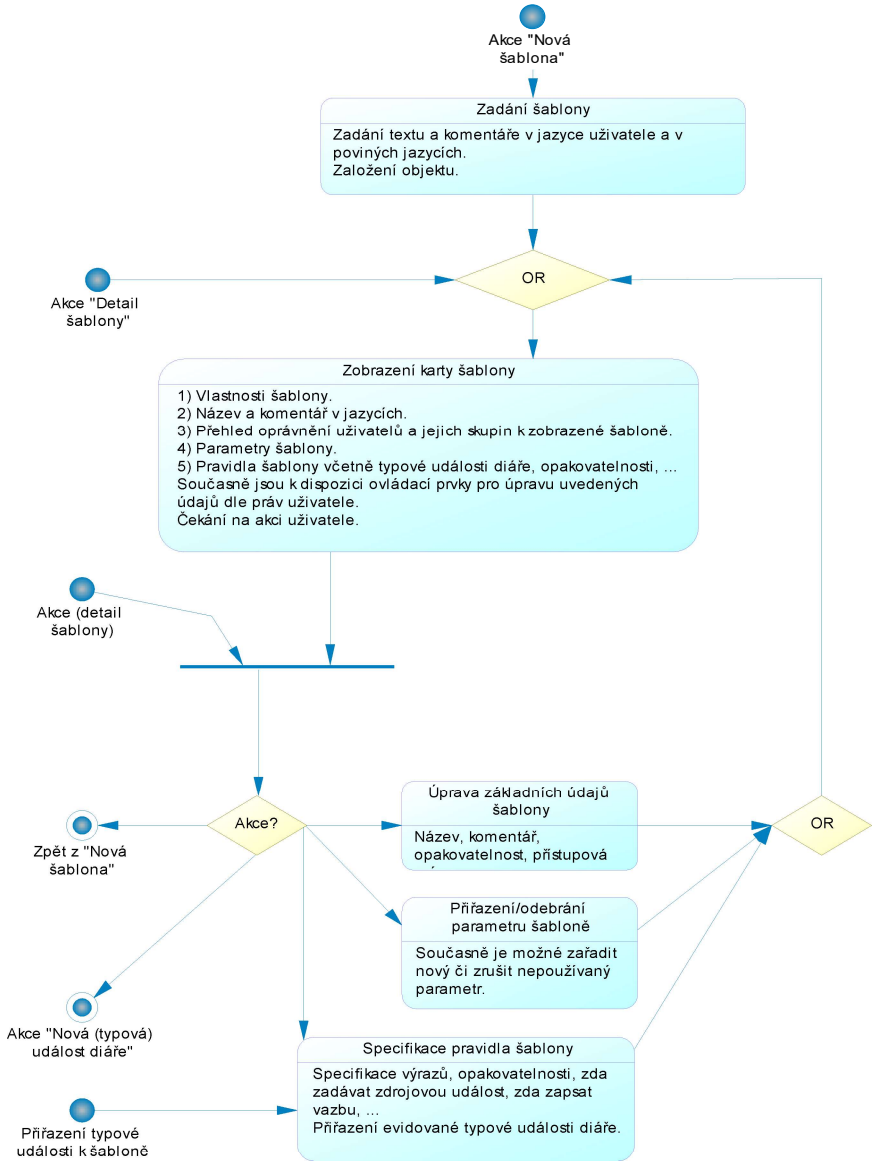
Obrázek 45- Správa šablony

Druhá část tohoto diagramu se věnuje jedné konkrétní šabloně, buď nové anebo stávající. V případě akce založení nové šablony se nejprve zadají její základní údaje (název a komentář) a je založena v datech (včetně záznamu v tabulce odpovídající entitě Objekt). Další postup zpracování akce Nová šablona je společný s akcí Detail šablony, tedy:

- Zobrazí se karta šablony, která obsahuje zejména tyto údaje:
  - Název v nastaveném jazyce.
  - Parametry šablony.
  - Pravidla šablony včetně jejich vlastností, tj. výrazy, opakovatelnost, typová událost diáře, …
  - Název a komentář ve všech jazycích, ve kterých jsou uvedeny.

- Přehled oprávnění uživatelů a jejich skupin k zobrazené šabloně.

- Dále se očekává některá z těchto akcí ze strany uživatele dle jeho oprávnění:
  - Skončit (vrátit se do přehledu šablon, kde již budou patrné provedené změny).
  - Úprava základních údajů šablony a to především název a komentář v určitém jazyce.
  - Přiřazení či případné odebrání parametru šablony.
  - Specifikovat pravidla šablony, včetně možnosti založení nové typové události diáře a jejího přiřazení aktuálnímu pravidlu šablony.
- V případě výběru jedné z výše uvedených akcí (kromě volby skončit) se po jejím ukončení opět zobrazí karta dané šablony včetně nově zadaných či pozměněných údajů. Opět se čeká na akci uživatele.



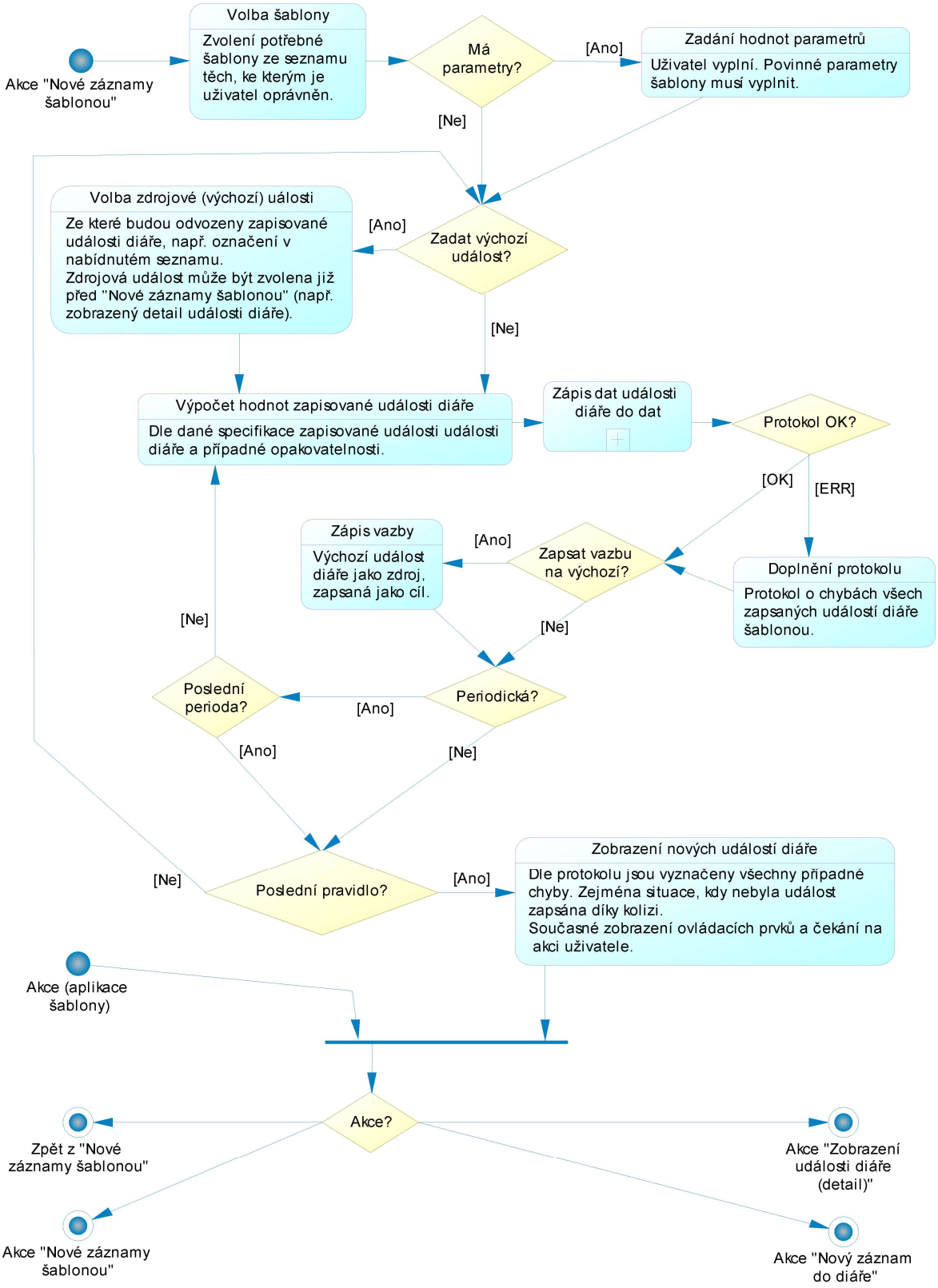
Obrázek 46- BPM detail a správa šablony

5.3.15 Diagram BPM – Aplikace šablon

Tento diagram se zaměřuje na specifikaci postupu při aplikaci šablony, tj. automatizované založení nových záznamů v diáři pomocí šablony. Specifikace šablon je popsána v kapitole „Diagram CDM – Šablony“.

Uživatel může iniciovat nové záznamy šablonou, události diáře, volbou relevantní položky menu, pak zpracování pokračuje těmito procesy:

- Volba šablony, kterou si uživatel dle svého oprávnění vybere ze seznamu již existujících šablon.
- Pokud uživatelem vybraná šablona má parametry, je uživatel vyzván k zadání jejich hodnot, v případě povinných je musí vyplnit.
- Dále v cyklu dle jednotlivých pravidel šablony:
    - Je-li nastaveno zadání zdrojové události, uživatel zadá zdrojovou událost, ze které budou odvozeny zapisované události.
    - Z výrazů se odvodí údaje zapisované události diáře.
    - Zapíše se událost diáře do dat, viz kapitola „Diagram BPM – Nová událost diáře“. V případě chyby, tj. kolize, se doplní protokol o zapsaných událostech šablonou.
    - V případě zadání zdrojové události je možné zapsat vazbu na původní zdrojovou událost, pokud to pravidlo specifikuje.
    - Pokud je událost vytvořená pravidlem šablony periodická a není to poslední perioda, znovu se z výrazů odvodí údaje zapisované události a událost diáře se zapíše.
- Pokud je vyhodnoceno poslední pravidlo šablony, zobrazí se seznam nově vytvořených událostí diáře danou šablonou včetně kolizí poznamenaných v protokolu a čeká se na akci uživatele:
    - Odejít z činnosti „Nové záznamy šablonou“ (ukončení této činnosti).
    - Zobrazit některé z nové založených událostí diáře (v detailu) s případnou jejich úpravou.
    - Založit nový záznam (událost diáře).
    - Vytvořit nové záznamy (události diáře) šablonou, tj. opakování tohoto postupu.



Obrázek 47- BPM aplikace šablon

6 Realizace řešení

Z pohledu této práce lze za realizaci informačního systému, respektive za základ jeho realizace, považovat fyzický datový model (PDM) a skript obsahující příkazy pro vytvoření struktury databáze, tj. tabulek, jejich sloupců a dalších databázových objektů, v systém řízení báze dat (DBMS) SAP Sybase SQL Anywhere 12.

Kompletní PDM a z něj generovaný skript odpovídající návrhu multilingválního diáře s hierarchií uživatelů jsou v elektronických přílohách této práce. V souvislosti s budoucí volbou programovacího jazyka, technické architektury, implementačního prostředí a případně i volbou jiné databáze může dojít ke zpřesnění PDM a tedy i ke změně skriptu pro vytvoření struktury databáze. Toto je jedním z cílů PD – podporovat celý životní cyklus IS.

Tato kapitola se zaměřuje hlavně na představení funkcí PD, jak:

- Odvodit PDM z CDM při udržení konzistence mezi těmito modely během celého životního cyklu budoucího informačního systému a jak PDM dále rozvíjet.
- Vygenerovat z PDM databázi, tj. skript s příkazy pro vytvoření databázových objektů.

6.1 Fyzický datový model (PDM)

V tomto případě jsou zde uspořádány tabulky, jejich sloupce a další objekty databáze budoucího IS, které souvisí s komplexním vedením diářů ve více jazycích včetně jejich sdílení pracovními týmy.

PDM lze v PD sestavit a aktualizovat těmito způsoby:

- Založením modelu a postupnou specifikací a propojováním jednotlivých objektů PDM. [10]
- Vygenerováním (Generate Physical Data Model) z konceptuálního nebo logického datového modelu, případně z objektového modelu. [10]
- Importem z existující databáze (Database Reverse Engineering, případně Update Model from Database). [10]

Zde se uplatnilo vygenerování PDM z navrženého a v předchozích kapitolách popsaného konceptuálního datového modelu (CDM). Vzhledem k tomu, že je CDM koncipován poměrně podrobně, tak je dosaženo vztahu mezi entitami a tabulkami 1:1, a tak bylo v PDM provedeno jen minimum úprav, které jsou popsány v následných kapitolách.

Tento model již plně odpovídá struktuře navrhované databáze a tak z něj lze vygenerovat skript pro její vytvoření, resp. pro vytvoření všech potřebných databázových objektů. Při vlastním programování této aplikace a zejména po rozhodnutí o architektuře IS (databázový server – aplikační server – klient), tedy při dalším postupu v realizaci IS, by mělo dojít ke zpřesnění a odpovídajícímu rozšíření PDM o uložené procedury, pohledy, události, webové služby, triggery, apod.

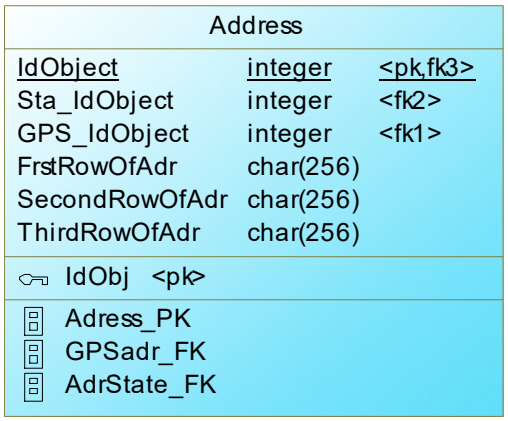
Pro dokreslení programátorských možností PD v PDM, jsou zde doplněny tyto typy objektů včetně případného programového kódu: Uložená procedura (Procedure), Událost (Event), Pohled (View), Uživatel (User) a Skupina uživatelů (Group) s nastavenými oprávněními.

Vzhledem k tomu, že informační schopnost budoucího systému je zřejmá již z předchozího CDM, tak se v této části této práce již neuvádí jednotlivé diagramy PDM. Ty jsou dostupné v přiložených modelech a jejich dokumentaci. Zde je uveden jen jeden diagram PDM jako příklad.

6.1.1 Objekty PDM

Jádrem fyzického datového modelu (PDM) jsou diagramy, ve kterých jsou tabulky propojeny referencemi, obvykle na bázi cizích klíčů. Pro zobrazení diagramů je nastavena (Model Option – Model Setting) notace „Conceptual“, která prezentuje diagramy, podobně jako jsou prezentovány diagramy v CDM. Pro zobrazování názvů objektů v diagramech je nastaveno (Model Option – Naming Convention) technické jméno (Code), které je také následně použito pro názvy databázových objektů. V diagramech se používají tyto grafické značky (Symbols) pro jednotlivé typy objektů:

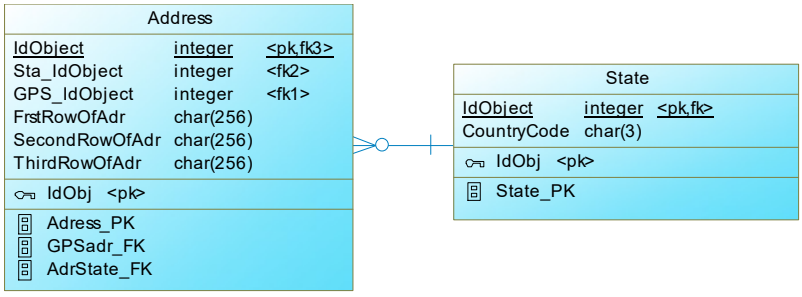
- Table – reprezentuje tabulku databáze, v jejíchž sloupcích budou uchovávány elementární informace. Pro diagramy PDM je v této práci nastaveno (Display Preferences), aby se zobrazoval tento obsah (Content): Název tabulky, všechny sloupce včetně datového typu a včetně informace, zda jsou součástí primárního nebo cizího klíče, všechny klíče i indexy. Oproti CDM je nastavena jiná barva výplně tohoto symbolu (Display Preferences – Format), aby se modely hned při prvním pohledu pohodlně rozlišily. Součástí specifikace tabulky je specifikace jejích sloupců, klíčů (primární, cizí, alternativní) a indexů. Dále lze specifikovat úrovně (alter, delete, insert, references, select a update) oprávnění přístupu k tabulce pro jednotlivé databázové uživatele (Users) i jejich skupiny (Groups). K dispozici je také možnost specifikovat související uložené procedury (Procedures), triggery (Triggers) a parametry (Physical Options). Tabulky je klíčovými objekty při generování databáze. [10]



Obrázek 48- Tabulka

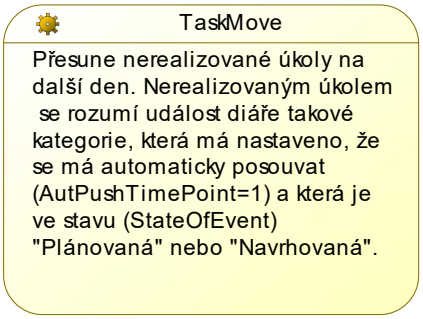
- Reference – reprezentuje propojení mezi dvěma tabulkami, mezi Parent Table („State“) a Child Table („Address“), které definuje jejich vzájemnou referenční integritu. Ve vlastnostech reference lze specifikovat propojení tabulek (Join) přímo uvedením sloupců jedné i druhé tabulky

anebo automatizovaně pomocí (obvykle primárního) klíče tabulky (Parent Table). Pokud jsou tabulky propojeny referencí, pak každá hodnota v relevantních sloupcích tabulky (Child Table) referuje ekvivalentní hodnotu ve sloupcích tabulky (Parent Table) – nelze tedy uvést hodnotu v Child Table, která by nebyla v Parent Table. Ve vlastnostech reference se dále specifikuje kardinalita a reakce v tabulce (Child Table) na změny primárního klíče v tabulce (Parent Table) – cascade, restrict, set null, set default. Reference se uplatní při generování databáze jako součást generování tabulky, na základě referencí se do databáze generují omezení (Constraint) tabulek, obvykle typu cizí klíč (foreign key). [10]



Obrázek 49- Ukázka reference mezi tabulkami

- Procedure – reprezentuje uloženou proceduru (stored procedure), případně funkci (function) tj. program určený pro vykonání určité sady příkazů a který může mimo jiné manipulovat s daty. Její zpracování může být řízeno parametry, se kterými je volána z aplikace. [10] Na záložce „Definition“ lze specifikovat kompletní programový kód procedury, případně za použití vzorových procedur (template). Dále lze specifikovat oprávnění používat proceduru jednotlivými databázovými uživateli (Users) i jejich skupinami (Groups). Procedura může být přiřazena k tabulce včetně případné vazby (Link/Extended Dependency), přičemž toto přiřazení je zaměřeno na zvýšení názornosti diagramu. Specifikace těchto procedur již úzce souvisí s programováním a v této práce je tedy definována jen jediná s cílem dokreslení možností PD. Procedury se uplatní při generování databáze.



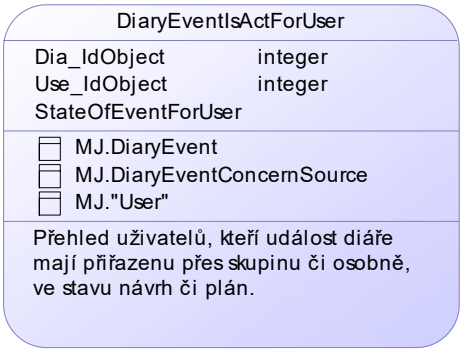
Obrázek 50- Procedura

- Link/Extended Dependency – reprezentuje vazbu mezi dvěma objekty PD libovolného typu, která upozorňuje na jejich vztah. [10] Tuto vazbu může uživatel definovat a aplikovat dle svých potřeb, případně je založena automaticky, např. v situaci, kdy PD v programovém kódu (Definition) uložené procedury automaticky identifikuje, kterých tabulek se týká a jaké další procedury či funkce volá, pak automaticky vytvoří vazby (typu Link/Extended Dependency) mezi danou procedurou a každou

Obrázek 51- Vazba

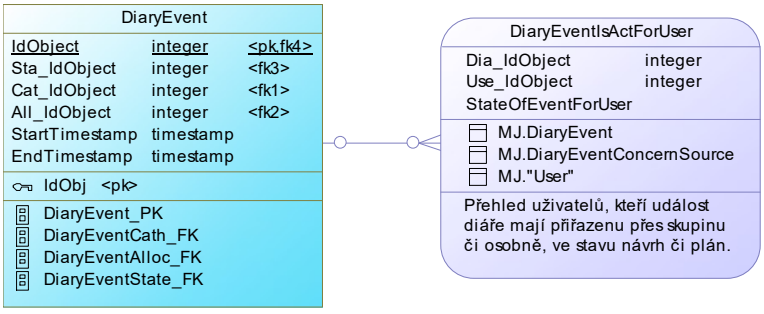
dotčenou tabulkou i volanou procedurou. Tyto vazby se neuplatní při generování databáze, jsou určeny jen pro vyšší srozumitelnost diagramu.

- View – reprezentuje databázový pohled, který poskytuje data ve stejné podobě jako tabulka. Na rozdíl od tabulky obsahuje pohled pouze předpis, jakým způsobem mají být data získána z tabulek a z jiných pohledů. [10] Tento předpis se definuje na záložce „SQL Query“ formou SQL dotazu. V rámci symbolu view se může zobrazovat seznam poskytovaných sloupců včetně jejich typu, indexů a seznam zdrojových tabulek a pohledů - tyto údaje jsou automaticky vytěženy z SQL dotazu. Dále lze specifikovat úroveň oprávnění (delete, insert, update, select) pracovat s pohledem pro jednotlivé databázové uživatele (Users) i jejich skupiny (Groups). Pohled může být ve vazbě s tabulkou či jiným pohledem (viz View Reference), přičemž toto přiřazení je zaměřeno na zvýšení názornosti diagramu. Specifikace pohledů již úzce souvisí s programováním. V této práce je tedy definován jen jeden s cílem dokreslení možností PD. Pohledy se uplatní při generování databáze.



Obrázek 52- Pohled

- View Reference – reprezentuje obdobné propojení jako reference, ale namísto vztahu dvou tabulek se jedná o vztah mezi pohledem a mezi tabulkou nebo druhým pohledem. View Reference se neuplatní při generování databáze, je určena zejména pro zvýšení srozumitelnost diagramu.



Obrázek 53- Propojení tabulky a pohledu

Dále jsou v této práci specifikovány objekty PDM korespondující s databázovými objekty, které se v diagramech zobrazují jen částečně v rámci symbolu tabulky (či pohledu) anebo se v nich nezobrazují vůbec:

- User – uživatel, tj. osoba, která má přístup (přihlášení či připojení) k databázi. Může být vlastníkem (Owner) databázových objektů. Může mít přiřazena oprávnění k systémovým funkcím databáze (Privilege) a k jednotlivým databázovým objektům (Permissions), která mají přednost před oprávněními zděděnými od skupin uživatelů, kterých je členem. Uživatelé se uplatní při generování databáze.
- Group – skupina (uživatelů) je určena ke zjednodušení přiřazování a správy oprávnění uživatelů, kteří mohou být do skupiny zařazeni. Pokud jsou jí přiřazena oprávnění (Privileges či

Permissions), tak je dědí uživatelé (Users), kteří jsou členy dané skupiny. Skupiny se uplatní při generování databáze.

- Domain – doména představuje uživatelsky specifikovaný nový datový typ (user-defined data type), který je založen na interním datovém typu (built-in data type) databáze a zpřesňuje ho například z hlediska délky, počtu desetinných míst, výčtem možných hodnot apod. Doménu lze používat jako datový typ pro sloupce tabulek. To umožní zpřesnění datových typů dotčených sloupců oproti použití interních datových typů, např. doména „Yes/No“ umožní v relevantních sloupcích jen hodnotu 1 nebo 2. Doména se může v diagramu zobrazovat v rámci symbolu tabulky jako datový typ daného sloupce. Domény se uplatní při generování databáze.
- Column – sloupec je vždy součástí tabulky a je nositelem její elementární informace. Sloupce se mohou v diagramu zobrazovat v rámci symbolu tabulky nebo symbolu pohledu. Ve specifikaci sloupce lze použít doménu jako datový typ, specifikovat výraz vypočítávaných (computed) sloupců a další omezení. Sloupce se uplatní při generování databáze jako součást generování tabulky. [10]
- Index – index reprezentuje datovou strukturu spojenou s tabulkou, která logicky uspořádává hodnoty klíčů. Umožňuje vyšší rychlost v přístupu k datům. Indexy se mohou v diagramu zobrazovat v rámci symbolu tabulky, kde jsou uvozeny ikonou indexu. Indexy se uplatní při generování databáze jako součást generování tabulky – to v této práci ale nebylo využito (viz následující kapitola). [10]
- Key – klíč je sloupec nebo kombinace sloupců, která unikátně identifikuje řádek tabulky. PDM podporuje primární, cizí a alternativní klíče. Klíče se mohou v diagramu zobrazovat v rámci symbolu tabulky, kde jsou uvozeny ikonou klíče. Klíče se uplatní při generování databáze jako součást generování tabulky. [10]
- Event – událost v systému řízení báze dat. Pokud nastane, může být automaticky aktivován určitý programový kód (např. uložená procedura). Událostí může být například: Start databáze, Deadlock apod. Událostí může být také situace, že nastal určitý čas, na který upozorní plánovač (scheduler). Součástí specifikace události je způsob spouštění (v případě aktivace časem se definuje i příslušný scheduler) a aktivovaný programový kód (handler). Události se v diagramu nezobrazují. Události se uplatní při generování databáze. [10]

K dispozici jsou v PDM i další typy objektů korespondující s databázovými objekty, které nejsou v této práci použity, ale není vyloučeno, že budou specifikovány při vlastním programování aplikace (např. trigger, web service, table space, sequence, login police,…), případně mohou být do modelu doplněny po jejich specifikaci v databázi (Update Model from Database).

6.1.2 Generování PDM z CDM

Konceptuální datový model v PD disponuje funkcí pro generování cílových fyzických datových modelů (menu Tools – Generate Physical Data Model). Základní vztah objektů PDM a CDM byl již zmíněn v kapitole „Konstrukce CDM s přihlédnutím ke generování PDM“. Pokud se generuje PDM poprvé, tak se specifikuje cílový model:

- Jméno, technické jméno a systém řízení báze dat (v tomto případě Sybase SQL Anywhere 12).
- Vlastnosti modelu (Model Options).

Generování je možné libovolně opakovat, přičemž je možné volit mezi úplným přepsáním již existujícího PDM a mezi spojením modelů, tedy integrací změn provedených v obou modelech (Preserve Modifications). V případě volby spojení modelů, jsou přehledně zobrazeny rozdíly mezi stávajícím PDM a tím, jak by vypadal PDM vygenerovaný právě z aktuálního CDM. Zde je možné o každém rozdílu rozhodnout, zejména zda:

- Nový objekt CDM se má vygenerovat i do PDM (implicitně nastaveno Ano).
- Změněná vlastnost objektu v CDM se má promítnout i do PDM (implicitně nastaveno Ano).
- Zda nový objekt v PDM má zůstat zachován (implicitně nastaveno Ano).
- Změněná vlastnost v PDM má zůstat zachována (implicitně nastaveno Ano).
- Zda má být změněné pořadí objektů z CDM promítnuto do PDM (implicitně nastaveno Ano).
- Zda má být změněné pořadí objektů v PDM zachováno (implicitně nastaveno Ano).

Integrace modelů umožňuje udržet trvalou konzistenci modelů během celého životního cyklu informačního systému. Aby toho bylo skutečně dosaženo, je nutné při generování PDM nastavit uchování závislostí mezi zdrojovými a cílovými objekty (Save Generation Dependencies).

Dále lze specifikovat, jak se mají stanovit jména těch objektů v PDM, které nemají svůj přímý zdrojový objekt v CDM, nýbrž vznikají automaticky při generování odvozením od objektů jiného typu. Jedná se například o specifikaci názvů automaticky doplňovaných sloupců tabulek, které jsou cizím klíčem apod.

6.1.3 Nastavení PDM

Objekty PDM byly z valné většiny založeny a nastaveny již jejich generováním z CDM, některé z nich byly v PDM upraveny. Jiné nově založeny a také byla provedena nastavení PDM:

- Nastaveno (viz Model Option), že se v diagramu mají namísto uživatelských názvů objektů (Name) zobrazovat jejich technické názvy (Code), které si uplatní i jako názvy databázových objektů.
- Upraven formát zobrazování některých symbolů v diagramu (např. barva výplně tabulky).

- Založen uživatel „Martin Janda“ s technickým názvem „MJ“. Tento uživatel nastaven jako implicitní uživatel (Default User) pro všechny typy objektů CDM, které tuto specifikaci umožňují (viz Model Option), a tedy i jako vlastník (owner) odpovídajících objektů databáze. Současně mu nastavena kompletní oprávnění k systémovým funkcím (Privileges). Oprávnění k objektům databáze (Permissions) vyplývá z toho, že je jejich vlastníkem.
- Založen uživatel „Tester“ s technickým názvem „TEST“. Nastavena mu některá oprávnění, zejména z důvodu prověření jejich projekce do databázových objektů.
- Založena skupina uživatelů „Uživatelé diáře“ s technickým názvem „DiaryUsers“ a nastavena některá oprávnění. Do ní zařazeni oba uvedení uživatelé.
- Založena uložená procedura „Posun úkolů na další den“ s technickým názvem „TaskMove“ a přiřazena k tabulce „DiaryEvent“, které se významně týká. Současně na záložce její definice (Definition) specifikován její programový kód zajišťující její funkcionalitu.
- Založena událost (Event) s technickým názvem „TaskMove“ aktivovaná časem. Současně na záložce „Sybase“ definován relevantní plánovač (Scheduler) „PredPulnoci“ specifikující, kdy se má tato událost spouštět, a spouštěný programový kód (Handler) volající stejnojmennou proceduru.
- Založen databázový pohled (View) „Událost je aktivní pro uživatele“ s technickým názvem „DiaryEventIsActForUser“, současně definován odpovídající dotaz (SQL Query), ze kterého si PD vyčte a prezentuje výstupní sloupce pohledu.
- Pro reference nastaveno (viz Model Option), že se mají do databáze promítat deklarativně, tj. pomocí specifikace omezení (add constraint … foreign key …) a nikoliv pomocí triggerů. Dále zde specifikovány implicitní akce (přednastaví se všem zakládaným referencím):
    - Cascade – změna primárního klíče v řádce rodičovské tabulky (Parent Table) se promítne do řádků tabulky (Child Table) změnou odpovídajícího cizího klíče.
    - Restrict – zrušení řádky v rodičovské tabulce není povoleno, pokud její primární klíč je v tabulce (Child Table) cizím klíčem.

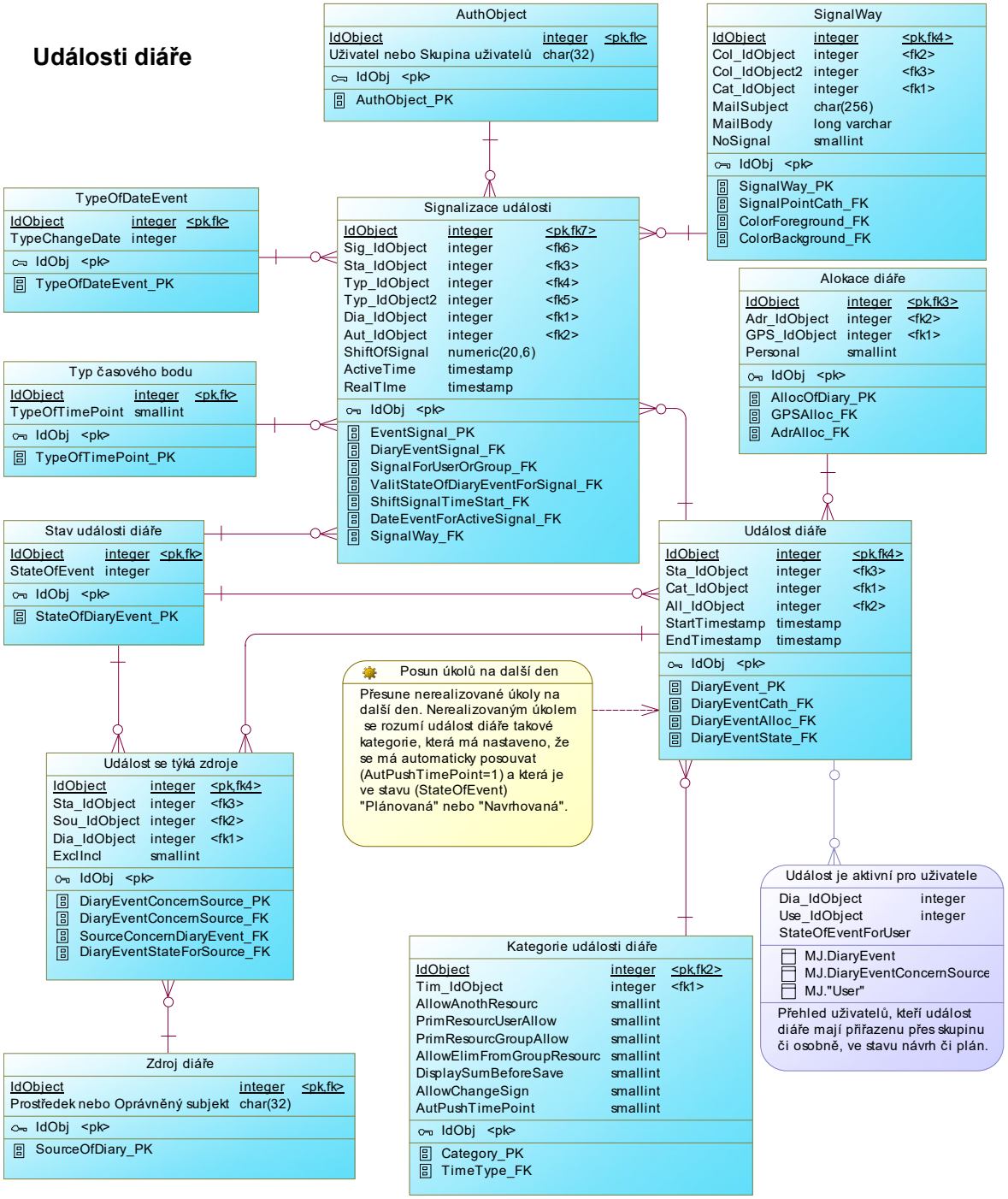
V několika konkrétních objektech PDM byly provedeny drobné úpravy oproti jejich vlastnostem převzatým z CDM (určitá korekce automatického generování). Jednalo se o tyto úpravy:

- V tabulce „User“ (entita „Uživatel“) provedeny tyto změny (nebylo možné ovlivnit při generování PDM z CDM, jak se přesně pojmenují objekty automaticky zakládané na základě inheritance), aby byla dodržena jednotnost pojmenování obdobných objektů v celém modelu:
    - Název sloupce „Id_Object“ opraven na „Jaz_Id_Object“.
    - Název sloupce „Opr_Id_Object“ opraven na „Id_Object“.

    - Názvy tří sloupců, každý představující cizí klíč odpovídající primárnímu klíči v tabulce „User“, opraveny na „Uži_Id_Object“, „Uži_Id_Object2“ a „Uži_Id_Object3“. Obdobně opraveny i jejich technické názvy (Code).
- S obdobným cílem upravena tabulka „RelatOfObj” (entita „Vazba objektů“) a související reference:
    - U reference s názvem „Vazba“ (Code „Link13“) upraveno propojení tabulek tím, že byl jako „Child Table Column” uveden sloupec „Id_Object“ (automaticky se změnily také indexy).
    - U reference s názvem „Předchůdce“ upraveno propojení tabulek tím, že byl jako „Child Table Column“ uveden sloupec „Obj_Id_Object“.
    - U klíče s názvem „IdObj“ nahrazen vygenerovaný sloupec sloupcem „Id_Object“.
    - Upraveno pořadí sloupců tabulky tak, aby sloupec odpovídající primárnímu klíči byl první.
- Tabulky „Comment“ (entita „Komentář v jazyce“) a „Text“ (entita „Text v jazyce“):
    - Do primárního klíče obou tabulek doplněn sloupec „IdObject“ a sloupec „Jaz_Id_Object“, současně změněno pořadí sloupců v primárním klíči.
- Tabulka „Object”:
    - Zrušena reference „Objekt reprezentující typ objektu“ (byla automaticky vygenerována, ale je fakticky duplicitní).
    - Zrušen index „ObjOfTypeObj_FK“ odpovídající zrušené referenci (díky které byl automaticky vygenerován).

6.1.4 Vzorový diagram PDM

Tento diagram, který je zde uveden jako příklad PDM diagramů, prezentuje klíčové tabulky modelu. Celý PDM včetně všech diagramů je k dispozici v přílohách této práce.



Obrázek 54- PDM události diáře

V tomto diagramu jsou patrné symboly:

- Tabulek propojených pomocí referencí. V rámci symbolů tabulek jsou také zobrazeny:

    - Sloupce tabulky, každý včetně datového typu a informace, zda je součástí primárního či cizího klíče.
    - Klíče (primární klíče).
    - Indexy, obvykle odpovídající primárnímu nebo cizímu klíči.
- Procedury propojené s tabulkou pomocí vazby „Link/Extended Dependency".
- Pohledu propojeného s tabulkou pomocí „View Reference". V rámci symbolu pohledu jsou také zobrazeny:
    - Poskytované výstupní sloupce.
    - Dotčené tabulky, ze kterých pohled čerpá poskytované informace.

Informační obsah prezentovaný tímto diagramem byl již popsán v odpovídajícím diagramu CDM, takže se zde zmiňují jen rozšíření diagramu PDM oproti zdrojovému diagramu CDM:

- Uložená procedura (Procedure) „TaskMove" posouvá datum nesplněných úkolů na další den. Je automaticky aktivována pomocí stejnojmenné databázové události (Event) řízené časovým plánovačem databáze (Scheduler).
- Pohled (View) „DiaryEventIsActForUser" poskytuje k dané události diáře přehled uživatelů, kterých se týká, a to i na základě členství uživatele ve skupinách uživatelů.

6.2 Generování databáze

V PDM disponuje PD funkcí pro generování databáze (menu Database -> Generate database), resp. databázových objektů na základě specifikace objektů PDM. Ke generování lze přistoupit dvěma způsoby (v rámci této práce byly testovány oba):

- První vygeneruje skript (RTF soubor) obsahující potřebné příkazy pro vytvoření databázových objektů. Ten se následně do databáze aplikuje z Interactive SQL volbou „Run Script". Tento skript je elektronickou přílohou této práce.
- Druhý přímo, tedy ihned, zakládá databázové objekty do databáze, se kterou se spojí pomocí ODBC (Open Database Connectivity).

V rámci dialogu po volbě generování databáze je možné specifikovat (záložka Options), které typy databázových objektů se mají vytvořit (tabulka, uživatel, …), které vlastnosti se jim mají nastavit (komentář, oprávnění,…) a jakým způsobem se mají založit (zrušit nejprve původní, …).

Také je možné vybrat (záložka Selections) jen konkrétní databázové objekty (konkrétní tabulky, pohledy, uživatele, procedury a domény), které se mají vytvořit. Výstupní skript může být uložen v jediném souboru anebo ve více souborech, pak je skript pro každý databázový objekt uložen v samostatném souboru. K dispozici je nahlédnutí skriptu (záložka Preview), kde je patrné, jakými příkazy a v jakém pořadí budou databázové objekty generovány.

K dispozici je také obrácená funkce (Update Model from Database), která umožní aktualizaci PDM ze specifikované databáze, ke které se PDM připojí pomocí ODBC. V tomto případě jsou do modelu vloženy všechny nebo jen vybrané databázové objekty (výběr objektů se provádí obdobně jako při generování databáze), případně jsou v modelu aktualizovány již existující objekty odpovídající databázovým objektům.

Základním východiskem generování databáze (také funkce Update Model from Database) je volba některého z definovaných DBMS, v rámci této práce byl zvolen „Sybase SQL Anywhere 12“.

Jaké databázové příkazy se mají vygenerovat ze specifikace jednotlivých typů objektů PDM lze pro daný DBMS uživatelsky definovat (viz menu „Database – Edit current DBMS“ anebo menu „Tool – Resources – DBMS Resource File Reference”). Zde jsou k dispozici rozsáhlé možnosti nastavení a přizpůsobení PDM ve vztahu k danému DBMS, případně i k novému DBMS.

Skripty definující generované databázové příkazy jsou zde definovány pro jednotlivé typy objektů PDM (např. tabulka, reference, …) a pro relevantní typy databázových příkazů (drop, create, …) zde nazývaných „Common object item” (položka). Skripty lze nalézt po této cestě „Root – Script – Object – [Typ objektu PD] – [Common object item]”. V rámci této práce byla ponechána jejich implicitní nastavení.

Objekty PDM, ze kterých se generují databázové objekty, disponují možností zobrazit odpovídající generované databázové příkazy na záložce „Preview“. V případě, že je generován z modelu jediný soubor pro všechny objekt, jsou tyto příkazy uspořádány do smysluplného pořadí.

V následných kapitolách je uvedeno, jaké databázové příkazy a na základě jakých položek skriptů (Common object item) jsou generovány z jednotlivých typů objektů PDM.

6.2.1 Skript – uživatel

Tento objekt PDM (User) má v „DBMS Resource File Reference” specifikovaný skript pro formulaci databázových příkazů v položkách (Common object item) „Drop“, „Create“ a „UserComment“, dále se uplatní položka „Create“ PDM objektu (Privilege) pro všechna uživateli přidělená oprávnění k systémovým funkcím. Na záložce „Preview“ objektů typu uživatel jsou zobrazeny generované databázové příkazy, například pro uživatele „TEST“:

```
if exists (select 1 from sys.sysuser where user_name = 'TEST') then
 drop user TEST
end if;
create user TEST IDENTIFIED BY test  force password change off;
comment on user TEST is 'heslo "test"';
grant backup,profile,validate to TEST;
```

Uživatel jména (User Id) „TEST“ nemusí měnit heslo po prvním přihlášení, má přidělené heslo „test“ a má přidělena oprávnění k systémovým funkcím. Tato oprávnění může mít přiřazena přímo anebo

je v PDM zdědit díky členství ve skupinách uživatelů. V tomto příkladu zdědil oprávnění k archivaci a validaci databáze od skupiny uživatelů „DiaryUsers“, naopak mu bylo explicitně odňato zděděné oprávnění „readfile“ a přímo mu bylo přiřazeno oprávnění k monitorování výkonu a diagnostice (profile).

6.2.2 Skript – skupina uživatelů

Tento objekt PDM (Group) má specifikovaný skript v položkách „Drop“, „Create“, „GroupComment“, „AfterCreate“, „Bind“ a dále se uplatní položka „Create“ PDM objektu (Privilege) pro všechna skupině přidělená oprávnění k systémovým funkcím. Na záložce „Preview“ uživatele jsou zobrazeny generované databázové příkazy, například pro skupinu „DiaryUsers“:

```
if exists (select 1 from sys.sysuser where user_name = 'DiaryUsers') then
        revoke connect from DiaryUsers
end if;
grant connect to DiaryUsers identified by DiaryUsers;
comment on user DiaryUsers is 'V této skupině jsou všichni uživatelé databáze s přístupem k IS Multilingvání diář. ';
grant group to DiaryUsers;
grant membership in group DiaryUsers to TEST;
grant membership in group DiaryUsers to MJ;
grant backup,readfile,validate to DiaryUsers;
```

Do skupiny „DiaryUsers“ jsou zařazeni uživatelé „MJ“ a „TEST“. Skupině jsou přiřazena oprávnění k archivaci a validaci databáze, dále ke specifikaci souboru jako zdroje dat pro příkaz SELECT (readfile).

6.2.3 Skript – doména

Tento objekt PDM (Domain) má specifikovaný skript v položkách „Create“ a „Drop“. Na záložce „Preview“ domény jsou zobrazeny generované databázové příkazy, například pro doménu „YesNo“:

```
if exists(select 1 from sys.sysusertype where type_name='YesNo') then
  drop domain YesNo
end if;
create domain YesNo as smallint not null check (@column in (1,2));
```

Součástí domény je specifikace omezení hodnot, v tomto případě na 1 nebo 2. Pokud je v databázi založen sloupec tabulky, který má jako datový typ doménu „YesNo“, pak se automaticky z domény převezme její „check constraint“ do jeho „column check constraint“.

6.2.4 Skript – tabulka

Pro tento objekt PDM (Table) se při generování databázových příkazů uplatní kromě specifikace vlastní tabulky i tyto související PDM objekty: Reference, Pkey (primární klíč), Column (sloupec) a Permission (oprávnění k tabulce). Jejich relevantní položky (Common object item) specifikované v „DBMS Resource File Reference” se uplatní v tomto pořadí (koresponduje s níže uvedeným příkladem):

- Reference – Drop: Zrušení cizích klíčů (v souvisejících tabulkách i v generované tabulce).
- Pkey – Drop: Zrušení primárního klíče.
- Table – Drop: Zrušení tabulky.
- Table – Create: Založení tabulky, s použitím (%TABLDEFN%) pro sloupce:
    - Column – Add: Přidání definice sloupce do definice tabulky.
- Table – TableComment: Komentování tabulky.
- Column – ColumnComment: Přidání komentáře ke každému sloupci tabulky.
- Pkey – Create: Založení primárního klíče.
- Pkey – PKeyComment: Komentování primárního klíče.
- Permission – Create: Přiřazení oprávnění k tabulce pro skupiny uživatelů a pro uživatele.
- Reference – Create: Založení cizích klíčů v souvisejících tabulkách a následně i v dané tabulce při současném jejich komentování (Reference – FKeyComment).

Na záložce „Preview“ tabulky jsou zobrazeny generované databázové příkazy, například pro tabulku „State“ se seznamem států používaných v adresách:

```
if exists(select 1 from sys.sysforeignkey where role='FK_ADRESS_ADRSTATE_STATE') then
   alter table MJ.Adress
      delete foreign key FK_ADRESS_ADRSTATE_STATE
end if;
if exists(select 1 from sys.sysforeignkey where role='FK_STATE_USERVALUE_OBJECT') then
   alter table MJ.State
      delete foreign key FK_STATE_USERVALUE_OBJECT
end if;
if exists(
  select 1 from sys.sysconstraint k
     join sys.systab t on (t.object_id = k.table_object_id and t.table_name='State')
     join sys.sysuserperms u on (u.user_id = t.creator and u.user_name='MJ')
  where
     k.constraint_type = 'P'
) then
   alter table MJ.State
  delete primary key
end if;
drop table if exists MJ.State;
create table MJ.State
(
  IdObject          IdObj                  not null,
  CountryCode       char(3)                    null
);
comment on table MJ.State is
'Číselník států, např. dle normy ISO 3166-1, tedy jakýkoliv stát na Zemi. Uplatní se při specifikaci adresy. ';
comment on column MJ.State.IdObject is
'Tento údaj je jednoznačným identifikátorem každého objektu. Každý záznam tedy je jednoznačně identifikovatelný pomocí identifikátoru „Id_Object“. Lze očekávat, že jeho hodnota bude v implementaci automaticky generována pomocí sequence.';
comment on column MJ.State.CountryCode is
'Třímistní kód státu dle normy ISO 3166-1.';
alter table MJ.State
  add constraint PK_STATE primary key clustered (IdObject);
```

```
grant SELECT on MJ.State to DiaryUsers;
grant INSERT,UPDATE on MJ.State to TEST;
alter table MJ.Adress
   add constraint FK_ADRESS_ADRSTATE_STATE foreign key (Sta_IdObject)
      references MJ.State (IdObject)
      on update cascade
      on delete restrict;
comment on foreign key MJ.Adress.FK_ADRESS_ADRSTATE_STATE is
'Adresa je umístěna v daném státu.';
alter table MJ.State
   add constraint FK_STATE_USERVALUE_OBJECT foreign key (IdObject)
      references MJ.Object (IdObject)
      on update cascade
      on delete restrict;
```

Jak je patrné, tak tabulka jako výchozí bod generování databázových objektů představuje poměrně komplexní PDM objekt. V rámci generování tabulek byla potlačena možnost generovat indexy korespondující s primárním a cizími klíči, neboť tyto indexy si databáze tvoří automaticky.

Příkazy související s primárním či cizím klíčem jsou k dispozici také samostatně u relevantního PDM objektu (Key) na záložce „Preview“.

6.2.5 Skript – pohled

Tento objekt PDM (View) má specifikovaný skript v položkách „Drop“, „Create“ a „ViewComment“, dále se uplatní položka „Create“ PDM objektu (Permission) pro všechna oprávnění uživatelů (User) a skupin uživatelů (Group) k danému pohledu. Na záložce „Preview“ pohledu jsou zobrazeny generované databázové příkazy, například pro pohled „DiaryEventIsActForUser“:

```
if exists(select 1 from sys.systable where table_name='DiaryEventIsActForUser' and table_type='VIEW' and
creator=user_id('MJ')) then
   drop view MJ.DiaryEventIsActForUser
end if;
create view MJ.DiaryEventIsActForUser  as
select distinct DiaryEvent.IdObject as Dia_IdObject, "User".IdObject as Use_IdObject, StateOfEventForUser
FROM MJ.DiaryEvent, MJ.DiaryEventConcernSource, MJ."User",
WHERE 1=1    /* tento dotaz je jen vzorový, bude nutno naprogramovat - patrně za použití rekurzivní SQL
funkce */
order by Dia_IdObject, Use_IdObject;
comment on view MJ.DiaryEventIsActForUser is
'Přehled uživatelů, kteří ji mají přiřazenu přes skupinu či osobně, ve stavu návrh či plán.';
grant SELECT on MJ.DiaryEventIsActForUser to DiaryUsers;
grant UPDATE on MJ.DiaryEventIsActForUser to TEST;
```

Tento pohled mohou používat pro získání dat (SELECT) všichni uživatelé, kteří jsou zařazeni do skupiny „DiaryUsers“ a dále uživatel „TEST“ může tento pohled modifikovat, to může implicitně i jeho vlastník (Owner).

6.2.6 Skript – událost

Tento objekt PDM (Event) má specifikovaný skript v položkách „Drop“, „Create“ a „Comment“. Na záložce „Preview“ události jsou zobrazeny generované databázové příkazy, například pro událost „TaskMove“:

```
drop event if exists MJ.TaskMove;
create event MJ.TaskMove schedule PredPulnoci between '23:50' and '23:55' every 24 hours start date '2016/01/01'
  enable
   at all
handler begin
  CALL MJ.TaskMove(current timestamp);
end;
comment on event TaskMove is
'Obvykle krátce před půlnocí každodenně aktivuje proceduru, která přesune nerealizované úkoly na další den.';
```

Jak je patrné, tak událost je spouštěna plánovačem (Scheduler) „PredPulnoci“, který je uvedeným příkazem také zakládán. Událost volá každodenně uloženou proceduru „TaskMove“.

6.2.7 Skript – uložená procedura

Tento objekt PDM (Procedure) má specifikovaný skript v položkách „Create“, „Drop“ a „Comment“, dále se uplatní položka „Create“ PDM objektu (Permission) pro všechna oprávnění uživatelů (User) a skupin uživatelů (Group) k dané proceduře. Na záložce „Preview“ události jsou zobrazeny generované databázové příkazy, například pro uloženou proceduru „TaskMove“:

```
drop procedure if exists MJ.TaskMove;
create procedure MJ.TaskMove (IN actTime timestamp)
begin
        UPDATE MJ.DiaryEvent
        SET DiaryEvent.StartTimestamp=DATEADD(day, 1, DiaryEvent.StartTimestamp),
                DiaryEvent.EndTimestamp=DATEADD(day, 1, DiaryEvent.EndTimestamp)
        FROM MJ.Category, MJ.StateOfDiaryEvent
        WHERE DiaryEvent.StartTimestamp<actTime
                and Category.IdObject=DiaryEvent.Cat_IdObject
                and category.AutPushTimePoint=1
                and StateOfDiaryEvent.IdObject=DiaryEvent.Sta_IdObject
                and StateOfDiaryEvent.StateOfEvent IN (1,2);
        commit;
end;
comment on procedure MJ.TaskMove is
'Přesune nerealizované úkoly na další den. Nerealizovaným úkolem se rozumí událost diáře takové kategorie, která má nastaveno, že se má automaticky posouvat (AutPushTimePoint=1) a která je ve stavu (StateOfEvent) "Plánovaná" nebo "Navrhovaná".';
grant EXECUTE on MJ.TaskMove to DiaryUsers;
grant EXECUTE on MJ.TaskMove to TEST;
```

Z příkladu je patrný i programový kód procedury. Proceduru mohou používat všichni uživatelé skupiny „DiaryUsers“, přičemž má uživatel „TEST“ navíc přímo specifikované oprávnění (tedy i v případě vyřazení z uvedené skupiny uživatelů bude oprávněn proceduru používat).

6.2.8 Skript – ostatní objekty

K dispozici je také možnost generovat databázové objekty či vlastnosti databázových objektů z dalších objektů PDM, které v této práci nejsou použity, a to: Business Role, Tablespace, Sequence, Web Service, Web Operation, Login policy, Mirror server, Text configuration atd., případně je možné tyto databázové objekty převzít do modelu z database (Update Model from Database).

6.2.9 Posloupnost příkazů ve skriptu vytvoření databáze

Pořadí příkazů ve vygenerovaném komplexním skriptu, tedy pořadí, v jakém se budou do databáze aplikovat, je toto:

- Zrušení událostí (drop event …).
- Zrušení triggerů (drop trigger …).
- Zrušení uložených procedur (drop procedure …).
- Zrušení cizích klíčů ve všech tabulkách (alter table … delete foreign key …).
- Zrušení pohledů (drop view …).
- Postupně pro každou tabulku:
    - Zrušení indexů (drop index …).
    - Zrušení primárních klíčů (alter table … delete primary key).
    - Zrušení tabulky (drop table …).
- Zrušení domén (drop domain …).
- Zrušení oprávnění pro skupiny, tj. group (revoke connect from …).
- Zrušení uživatelů (drop user …).
- Založení uživatelů (create user …) a nastavení jejich oprávnění k systémovým funkcím.
- Založení skupin (grand connect to …; grant group to …; grant membership in group …; grand profile …).
- Založení domén (create domain …).
- Založení tabulek (create table …) včetně primárních klíčů (alter table … add constraint … primary key) a indexů (create index … on … (… ASC)).
- Založení pohledů (create view … as select …).
- Doplnění specifikace cizích klíčů (alter table … add constraint … foreign key).
- Založení uložených procedur (create procedure …).
- Založení triggerů (create trigger … on …).
- Založení událostí (create event …).

6.2.10 Ověření funkčnosti generování databáze

Pro ověření generovaného skriptu byla založena databáze se jménem „DiaryDatabaseName“ v systému řízení báze dat Sybase SQL Anywhere 12. Současně je definováno odpovídající ODBC „DiaryODBCName32“.

Pro generování skriptu byla zvolena varianta do jediného souboru, což je pak značně pohodlnější při vlastním spouštění skriptů. Identifikovány dva drobné problémy komplexního skriptu při aplikaci na databázi a ty byly vyřešeny, jednalo se o:

- V PDM bylo nastaveno pro jednotlivé databázové objekty, že je vlastní uživatel „MJ“, tedy i ve skriptu se tento uživatel objevuje v jejich prefixu, a to i v úvodních fázích skriptu, např. „drop event if exists MJ.TaskMove;“, kdy ještě uživatel „MJ“ není v databázi založen. Vyřešeno tak, že při prvním generování skriptu zvolen jen typ databázového objektu „User“ a jen konkrétní uživatel „MJ“ – tento skript byl na databázi aplikován (Run script v ISQL), tedy uživatel „MJ“ založen včetně jeho privilegií.
- Pokud je v tabulce PDM definován primární nebo cizí klíč, vede to ve skriptu na odpovídající databázové příkazy pro jeho založení (alter table … add constraint … primary/foreign key). Databáze navíc automaticky sama zakládá odpovídající index, tedy by se jednou zakládal díky funkcionalitě databáze a podruhé duplicitně díky specifikaci v PDM. To je vyřešeno tak, že byly z generování vyloučeny PDM objekty reprezentující indexy odpovídající primárním a cizím klíčům.

S takto specifikovanými parametry vygenerovaný výstupní skript byl bezchybně aplikován a vytvořil všechny specifikované databázové objekty. Též bylo prakticky prověřeno přímé generování databáze, ke které se PD připojil pomocí ODBC. V příloze této práce je uveden zde popsaný skript pro vytvoření databázových objektů.

7 Závěr

Tato práce se věnuje analýze a návrhu informačního systému „multilingvální diář s hierarchií uživatelů“. Jejím hlavním cílem bylo vytvoření konceptuálního datového a procesního modelu, které reprezentují návrh uvedeného IS. Z konceptuálního datového modelu byl vygenerován fyzický datový model a z něho následně skript obsahující příkazy pro vytvoření struktury databáze, tj. tabulek, jejich sloupců a dalších databázových objektů, v systém řízení báze dat (DBMS) SAP Sybase SQL Anywhere 12.

V teoretické časti, byl obecně popsán vývoj informačního systému a PowerDesigner (komplexní modelovací nástroj softwarového inženýrství (CASE) zaměřený na návrh informačních systémů i na jejich rozvoj). Dále byla popsána analýza požadavků na informační systém.

Poznatky z teoretické části byly následně použity pro vytvoření konceptuálního modelu. Tento model se systematicky testuje pomocí procesního modelu, což vede na změny v jeho návrhu, takže se zpřesňuje.

Na základě konceptuálního modelu byl vygenerován v PowerDesigneru fyzický datový model, kde bylo potřeba provést minimum úprav, aby mohl být vygenerován skript pro vytvoření databáze. Pro ověření generovaného skriptu byla založena databáze se jménem „DiaryDatabaseName“ v systému řízení báze dat Sybase SQL Anywhere 12.

Po vyřešení identifikovaných drobných problémů komplexního skriptu při aplikaci na databázi, byl vygenerovaný výstupní skript bezchybně aplikován a vytvořil všechny specifikované databázové objekty. Též bylo prakticky prověřeno přímé generování databáze, ke které se PD připojil pomocí ODBC. Bylo dosaženo cílů specifikovaných v zadání této práce.

Tato práce pro mě byla velmi zajímavá s ohledem na vyzkoušení kompaktního vývojového prostředí PowerDesigner, kde jsem si osvojit především konceptuální, procesní a fyzické modelování.

8 Přílohy

CDM - elektronická

BPM - elektronická

PDM - elektronická

Skript - elektronická

Dokumentace generovaná z modelů - elektronická

9 Seznam obrázků

10 Použité zkratky

BPM - business proces model

CASE - Computer-aided software engineering

CDM - konceptuální datový model

DBMS - database management system (systém řízení báze dat)

HTML - HyperText Markup Language

IDM - ideální datový model reprezentující požadovanou informační schopnost IS

IS - informační systém

MDE - Model-driven engineering

ODBC - Open Database Connectivity

PD - PowerDesigner

PDM - fyzický datový model

RTF - rich text format

SQL - system query language (obvykle se objevuje v názvu DBMS)

UML - Unified Modeling Language

WBS - work breakdown structure

GPS - global positioning system

11 Použité zdroje

[1] Time management. In: *Wikipedia.org* [online]. 2015 [cit. 2015-11-5]. Dostupné z: https://cs.wikipedia.org/wiki/Time_management

[2] Ganttův diagram. In: *Wikipedia.org* [online]. 2015 [cit. 2015-11-5]. Dostupné z: https://cs.wikipedia.org/wiki/Gantt%C5%AFv_diagram

[3] Řízení projektů. In: *Wikipedia.org* [online]. 2016 [cit. 2016-01-15]. Dostupné z: https://cs.wikipedia.org/wiki/%C5%98%C3%ADzen%C3%AD_projekt%C5%AF

[4] ŘEPA, Václav. *Analýza a návrh informačních systémů*. Vyd. 1. Praha: Ekopress, 1999. ISBN 80-86119-13-0.

[5]MOLNÁR, Zdeněk. *Efektivnost informačních systémů*. 1. vyd. Praha: Grada, 2000. Systémová integrace. ISBN 80-7169-410-X.

[6] BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: podnik v informační společnosti*. 3., aktualiz. a dopl. vyd. Praha: Grada, 2012. Management v informační společnosti. ISBN 978-80-247-4307-3.

[7] POLÁK, Jiří, Vojtěch MERUNKA a Antonín CARDA. *Umění systémového návrhu: objektově orientovaná tvorba informačních systémů pomocí původní metody BORM*. 1. vyd. Praha: Grada, c2003. ISBN 80-247-0424-2.

[8] ŘEPA, Václav. *Podnikové procesy: procesní řízení a modelování*. 2., aktualiz. a rozš. vyd. Praha: Grada, 2007. Management v informační společnosti. ISBN 978-80-247-2252-8.

[9] CONOLLY, Thomas, Carolyn E BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Vyd. 1. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.

[10] Data modeling: PowerDesigner. *www.sybase.com* [online]. 2015 [cit. 2016-04-13]. Dostupné z: http://infocenter.sybase.com/archive/topic/com.sybase.infocenter.dc38058.1530/doc/pdf/data_modeling.pdf

[11] *PowerDesigner: Conceptual data model* [online]. 2004 [cit. 2016-04-13]. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.199.8595&rep=rep1&type=pdf

[12] *Business process modeling: PowerDesigner* [online]. 2011 [cit. 2016-04-13]. Dostupné z: http://infocenter.sybase.com/help/topic/com.sybase.infocenter.dc38088.1610/doc/pdf/business_process_modeling.pdf

[13] PowerDesigner. *Wikipedia.org* [online]. 2016 [cit. 2016-04-12]. Dostupné z: https://en.wikipedia.org/wiki/PowerDesigner

[14] SQL_Anywhere. *En.wikipedia.org* [online]. 2015 [cit. 2016-04-13]. Dostupné z: https://en.wikipedia.org/wiki/SQL_Anywhere