

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Student : **Jakub Sychra**
Vedoucí bakalářské práce : **RNDr. Helena Palovská, PhD.**
Recenzent bakalářské práce : **Ing. Jan Klas, PhD.**

TÉMA BAKALÁŘSKÉ PRÁCE

**Návrhové vzory pro databáze v hospodářských
organizacích**

ROK : 2008

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a že jsem uvedl všechny použité prameny a literaturu, ze kterých jsem čerpal.

V Praze dne 1.1.2008

.....
podpis

Chtěl bych velmi poděkovat RNDr. Heleně Palovské, PhD. za cenné rady a odborné vedení bakalářské práce.

Abstrakt

Při navrhování databáze se často setkáváme s případy, kdy je potřeba zachytit nějakou situaci z reality, která se často opakuje a její řešení není zcela jednoduché. Cílem této práce je popis těchto situací pomocí datových modelů. Ty představují možný způsob řešení, které lze považovat za vzorové a aplikovatelné i pro jiné situace.

Úvodní kapitola má za úkol uvést čtenáře do problematiky návrhových vzorů. Ty jsou v informatice úspěšně využívány při návrhu objektově orientovaných programů. Pro úspěšné pochopení modelů zobrazených v této práci je druhá kapitola věnována základům datového modelování. Zbývající kapitoly se věnují tvorbě návrhových vzorů databáze hospodářské organizace. Každá z kapitol je zaměřena na některou část z databáze a popisuje její nejdůležitější entity a vazby.

V závěru jsou uvedeny příklady modelů databáze, ve kterých jsou použity poznatky vycházející z této práce.

Abstract

While designing the database we may often face a problem with capturing of certain situation from reality, that is revealed quite often and its solution is far from elementary. The aim of this work is to describe these situations in terms of database models. These models represent method that can be used as a pattern and is even applicable to similar situations.

The first chapter is here as an introduction to design patterns. Design patterns are successfully used in design of object oriented software. Second chapter is dedicated to the basics of the data modeling to gain the ability to read the models presented in this work. The remaining chapters are dedicated to creation of design patterns for enterprise database. Each chapter is dedicated to one of the database parts and describes the key entities and their relationships.

There are some examples of database models in appendix of this thesis, that include some knowledge coming out from this work.

Obsah

ABSTRAKT	4
ABSTRACT	5
OBSAH	6
ÚVOD	8
1. CO JSOU NÁVRHOVÉ VZORY	9
2. TEORETICKÉ ZÁKLADY DATOVÉHO MODELOVÁNÍ.....	10
2.1. ENTITA	10
2.3. RELACE	11
2.3.1. Relace 1:1	12
2.3.2. Relace 1:N	12
2.3.3. Relace N:N	13
3. PODNIK A JEHO OKOLÍ.....	15
3.1. LIDÉ	15
3.2. ZAMĚSTNANCI.....	17
3.3. ORGANIZACE PODNIKU	18
4. OBJEKTY V PODNIKU	23
4.1. ROZDĚLENÍ VÝROBKŮ	23
4.2. STRUKTURA OBJEKTŮ.....	24
5. ČINNOSTI V PODNIKU	26
5.1. DĚLENÍ ČINNOSTÍ	26
5.2. VYUŽITÍ LIDSKÝCH ZDROJŮ V ČINNOSTECH	28
5.3. VYUŽITÍ OBJEKTŮ V ČINNOSTECH.....	30
6. OBCHODOVÁNÍ	34
6.1. SMLOUVY	34
6.2. PODNIKOVÉ DOKUMENTY	35
6.3. OBCHODY V CIZÍ MĚNĚ.....	40
7. UNIVERZÁLNÍ MODEL.....	43
ZÁVĚR.....	44
PŘÍLOHY.....	45

PŘÍLOHA A: KNIHA ÚKOLŮ	46
PŘÍLOHA B: ŠKOLENÍ A SEMINÁŘE	48
POUŽITÁ LITERATURA.....	49
ELEKTRONICKÉ ZDROJE.....	49
TERMINOLOGICKÝ SLOVNÍK	50

Úvod

Termín návrhový vzor se dostal do širšího povědomí ve druhé polovině 70. let dvacátého století. Zasloužil se o to americký architekt rakouského původu Christopher Alexander svými publikacemi o uplatnění těchto vzorů v architektuře.

Přestože byly návrhové vzory používány původně v architektuře a stavitelství, je jejich pole působnosti daleko širší a lze je aplikovat v mnoha oborech lidské činnosti. Pomineme-li již zmiňovanou činnost stavitelskou, použití vzorů je patrné při konstrukci automobilů a jejich částí, komponování hudby i v samotném lidském chování.

V informační společnosti začaly hrát návrhové vzory důležitou roli ve druhé polovině let devadesátých. Našly své uplatnění při navrhování objektově orientovaného softwaru.

Cílem této práce je prozkoumat možnosti použití návrhových vzorů v jiné oblasti počítačové vědy, a to při navrhování databáze.

První kapitola je zaměřena na problematiku návrhových vzorů. O těch se v souvislosti s databázemi příliš často nemluví. Jsou zde popsány společné rysy návrhových vzorů v různých aplikačních oblastech.

Následující kapitola se věnuje základům datového modelování. V úvodu je představen formát v jakém jsou datové modely následně vytvářeny. Součástí této kapitoly je představení jednotlivých konstrukčních prvků diagramu datového modelu. Znalost základů datového modelování je předpokladem pro pochopení a orientaci v datových modelech.

Obsahem zbývajících kapitol je analýza různých oblastí uvnitř podniku, které by měla databáze podnikového informačního systému pokrývat. Každá kapitola se zaměřuje na jinou oblast. V úvodu jednotlivých kapitol jsou představeny základní entity a vazby mezi nimi, s jejichž pomocí lze řešit některé situace. Jsou zde zmíněny i takové případy, pro které stávající řešení není použitelné, a tento nedostatek je odstraněn dalšími úpravami. Výsledky analýzy třetí a čtvrté kapitoly jsou následně použity v kapitolách následujících.

Seznam odkazů na použitou literaturu je uveden na konci jednotlivých oddílů.

1. Co jsou návrhové vzory

Přestože se historie návrhových vzorů začala psát již před 30 lety, neexistuje pro ně žádná oficiální definice. Nejlépe vzory popisují slova jejich průkopníka Christophera Alexandra: „Každý vzor popisuje problém, který se v našem prostředí opakovaně vyskytuje. Poté popisuje jádro řešení daného problému tak, že nám umožňuje toto řešení mnohokrát použít.“ Autor měl sice na mysli architektonický vzor, ale tuto definici lze aplikovat v každé oblasti využívající vzorů.

Výhody použití návrhových vzorů:

- Návrhový vzor může urychlit proces vývoje použitím vyzkoušených, osvědčených příkladů.
- Užití návrhových vzorů umožňuje předcházet problémům při řešení složitějších situací.

V oblasti návrhu objektově orientovaného softwaru je použití návrhových vzorů omezeno na řešení netriviálních a opakujících se situací. Malé problémy se často v praxi vyskytují v různých obměnách. Proto je jejich individuální řešení pomocí znalostí a schopností specialisty efektivnější než řešení pomocí návrhových vzorů. Použití vzorů pro často se opakující situace zvyšuje pravděpodobnost jejich znovupoužitelnosti.

Každý návrhový vzor, bez ohledu na oblast ve které je použit, je výsledkem kombinace praktických a teoretických zkušeností.

[2] [3] [7]

2. Teoretické základy datového modelování

Jednou z vlastností návrhových vzorů používaných v objektově orientovaném přístupu programování, je jejich nezávislost na vývojovém prostředí, ve kterém je prováděna implementace. Při zkoumání vzorů pro databázi podniku jsou v této práci použity datové modely na konceptuální úrovni. Tato úroveň má za úkol věrně zachytit realitu z pohledu informačního světa bez ohledu na technologické a informatické prostředí.

Pro datové modelování je v praxi užíváno hned několik přístupů. Vzájemně se od sebe liší především použitými grafickými symboly, nicméně ve výsledku je lze použít pro většinu situací. V této práci je pro grafické znázornění datových modelů použito diagramu entit a vztahů (dále jen ERD), a to v Bachmanově notaci.

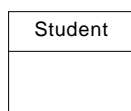
Zbývajících část této kapitoly je věnována jednotlivým konstrukčním prvkům ERD, jejichž představení je důležité pro pochopení a orientaci v datových modelech, které jsou v této práci používány. Mezi tyto prvky patří entita, atribut a relace.

[2] [9] [10]

2.1. Entita

Tento klíčový element, bez něhož se žádný datový model neobejde, představuje datový typ objektů, o nichž je potřeba uchovávat nějaké informace. Jsou to prvky jak fyzické (osoba, věc), tak i prvky povahy nehmotné (pozice, transakce). Entitnímu typu zpravidla odpovídá tabulka v databázovém relačním modelu. Výskyt entity (instance) pak představuje řádek v této tabulce.

Pomocí ERD interpretujeme entitu jako čtyřúhelník, v jehož horní části je uveden její název. Ten je vždy uveden jako podstatné jméno v čísle jednotném. Na obr. 2.1 je uveden příklad entity s názvem Student.



Obr. 2.1

[1] [9]

2.2. Atribut

Jednotlivé výskyty entity se od sebe liší svými vlastnostmi. U osob se může jednat o jméno, rodné číslo, atd. Tyto vlastnosti se v datovém modelování označují jako atributy a v diagramu je zapisujeme pod název entity.

Atributy lze rozdělit do dvou základních skupin – identifikátory a deskriptory. Identifikátor je vlastnost, jejíž hodnotu má každý výskyt dané entity jedinečnou a jiný výskyt této entity již této hodnoty nabývat nemůže. V evidenci studentů VŠE lze za takovýto identifikátor považovat *xName*. Naproti tomu jméno je vlastnost osoby, jejíž hodnoty se mohou u některých výskytů shodovat. Takovýto atribut se nazývá deskriptor.

Další rozdělení atributů vychází z toho, zda je hodnota pro jednotlivé instance povinná, či ne. Pro výskyt entity osoba budou atributy *DatumNarozeni* a *Jmeno* pravděpodobně povinné, na rozdíl například od atributu *Telefon*. První dva jmenované atributy patří do skupiny totálních atributů. Atribut *Telefon* je zástupce skupiny parciálních atributů. Způsob zápisu atributů do modelu je uveden na obr. 2.2. Pro identifikátor je uvedeno označení <pi>. Dále je uveden datový typ pro hodnoty každého atributu a příznak <M> (z anglického slova mandatory) v případě, že je povinný.

Student			
<u>xName</u>	<pi>	VA50	<M>
Jmeno		VA50	<M>
DatumNarozeni		D	<M>
Telefon		VA20	

Obr. 2.2

[9]

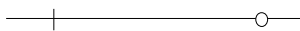
2.3. Relace

Termínem relace označujeme vztah mezi jednotlivými entitami. Nejčastěji se lze setkat s relacemi mezi dvěma entitami, tzv. binárními relacemi. Některé situace je nutné popsat více než dvěma entitami. O takovýchto vzájemných vztazích mezi několika entitami hovoříme jako o polyárních relacích.

U relací sledujeme následující dvě charakteristiky – volitelnost a kardinalitu.

Volitelnost relace poskytuje informaci o tom, zda je existence výskytu entity závislá na existenci výskytu druhé entity ze vztahu. Pokud se instance entity A vyskytuje u každého výskytu entity B, pak hovoříme o povinném (totálním) vztahu. V opačném případě se jedná o vztah volitelný (parciální). Vztah je v diagramu ERD znázorněn pomocí čar mezi entitami.

Volitelnost vztahu se pak znázorňuje jak je uvedeno na obrázku 2.3. Levá strana (svislá čárka) představuje povinnou část vztahu. Na druhé straně je pak volitelná část vztahu označená prázdným kolečkem.



Obr. 2.3

Naproti tomu kardinalita představuje poměr počtu výskytů entity na jedné straně vztahu k počtu výskytů entity na opačné straně vztahu. Kardinalita nabývá pouze dvou hodnot a to 1 a N. Z těchto dvou hodnot mohou vzniknout následující tři kombinace.

2.3.1. Relace 1:1

Jeden výskyt entity A vstupuje do vztahu právě s jedním výskytem entity B. Obrázek 2.4 znázorňuje situaci, kdy mají zaměstnanci přidělen jeden služební automobil. K zaměstnanci je vztah povinný, tzn. že automobil musí být vždy k někomu přidělen. Ne každý zaměstnanec má však automobil k dispozici. Tuto volitelnost znázorňuje druhá část vztahu.



Obr. 2.4

2.3.2. Relace 1:N

Jeden výskyt entity A vstupuje do vztahu s více instancemi entity B. Zároveň však jedné instanci entity B lze přidělit pouze jednu instanci entity A. Se vztahy typu 1:N se lze v realitě setkat nejčastěji. Příkladem vztahu tohoto typu může být situace z obrázku 2.5. V tomto případě jsou zaměstnanci rozmístěni na pracovišti do kanceláří, přičemž v jedné kanceláři může být více zaměstnanců. Každý zaměstnanec musí mít kancelář. Některé kanceláře mohou zůstat nevyužité.



Obr. 2.5

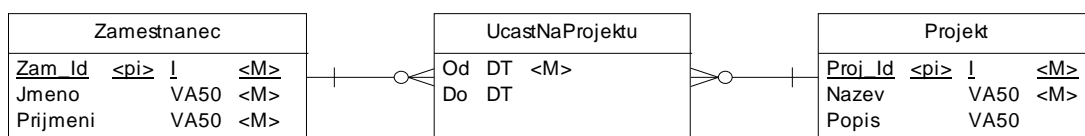
2.3.3. Relace N:N

Jeden výskyt entity A vstupuje do vztahu s více instancemi entity B. Zároveň jeden výskyt entity B vstupuje do vztahu s více objekty entity A. Model na obrázku 2.6 popisuje přidělování zaměstnanců k jednotlivým projektům. Každý zaměstnanec se může podílet na několika projektech. Přičemž na velkých projektech se podílí více zaměstnanců.



Obr. 2.6

Model typu M:N lze dekomponovat na dva vztahy 1:N. Tento přístup lze použít v případě, kdy je potřeba uchovávat o vztahu nějaké informace. Model na obrázku 2.7 umožňuje sledovat účast zaměstnanců na jednotlivých projektech a uchovávat historii těchto účastí. Vztahy jsou k entitám *Zamestnanec* a *Projekt* povinné. Účast na projektech je podmíněna existencí projektu i zaměstnance. K entitě *UcastNaProjektu* jsou vztahy volitelné. Ne každý projekt musí být zpracováván a někteří zaměstnanci nejsou zapojeni do žádného projektu, např. noví zaměstnanci. Jednotlivé výskyty entity *UcastNaProjektu* nemá smysl označovat identifikátorem.



Obr. 2.7

V dalším textu je na vztahy typu N:M nahlíženo z obou pohledů. Obecně je dekompozice vztahu uplatněna v případě, kdy je nutné u vztahu sledovat některé další náležitosti. V ostatních případech je použito jediného vztahu jako na obrázku 2.6.

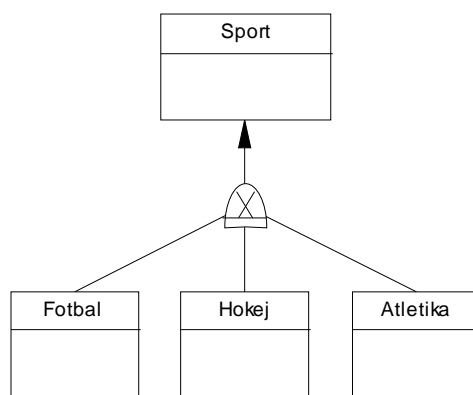
[8] [10]

2.4. Generalizace – specializace

Generalizace – specializace představuje speciální typ vztahu, kdy entity nižší úrovně jsou závislé na jiné entitě z vyšší úrovně. Entitu podle úrovně nazýváme podtypem nebo nadtypem. Při generalizaci se ze dvou a více konkrétních entit vytváří jedna abstraktnější. Specializací je pak rozuměn opačný proces. Pro vztahy generalizace – specializace se někdy používá označení ISA hierarchie. Mezi charakteristické znaky pro tento typ vztahu patří:

- každý výskyt podtypu je výskytem nadtypu
- každý výskyt podtypu přebírá všechny atributy nadtypu

Způsob interpretace ISA hierarchie v ER diagramu je znázorněn na obrázku 2.8.



Obr. 2.8

Pro detailnější popis vztahů mezi podtypy se někdy používá členění na exkluzivní a inkluzivní podtypy. Do skupiny exkluzivních podtypů patří ty podtypy, které se vzájemně nepřekrývají. Tedy výskyt nadtypu má maximálně jeden výskyt v některém z podtypů. Pripustíme-li možnost, že výskyt nadtypu je členem více podtypů, patří pak podtypy z této skupiny mezi inkluzivní podtypy.

Další způsob členění vychází z toho, zda každý výskyt nadtypu musí být výskytem jednoho z podtypů v rámci jedné skupiny. Patří-li každý výskyt hlavní entity alespoň do jednoho podtypu, hovoříme o kompletním pokrytí nadtypu. V opačném případě, stavu, kdy výskyt nadtypu nemá své zastoupení v žádném z podtypů, říkáme částečné pokrytí.

[8] [10]

3. Podnik a jeho okolí

Podnik dosahuje svých cílů prostřednictvím lidí, kteří svým úsilím chod tohoto podniku zajišťují. Proto se ve většině datových modelů pro podnikové informační systémy objevuje entita, která má osoby z reality reprezentovat.

3.1. Lidé

Nejprve je nutné určit způsob interpretace osoby v datovém modelu a její náležitosti. Entitu, která bude zastupovat osoby uvnitř podniku nazveme *Osoba*. Jak již bylo uvedeno v předchozí části, aby bylo možné jednotlivé osoby odlišit a vyhnout se tak jejich případné záměně, je nutné přidělit každému novému záznamu jednoznačný identifikátor. Přestože se v případě evidence osob takový identifikátor již nabízí v podobě rodného čísla, lepší volbu představuje vytvoření nového identifikátoru. Případy duplicitních rodných čísel a jejich změny během života jedince jsou velmi vzácné. Nicméně následné změny v databázi mohou způsobit nemalé komplikace, proto rodné číslo zařadíme mezi deskriptory. K identifikaci osoby použijeme uměle vytvořený identifikátor *Os_Id*. K výše uvedeným atributům lze přiřadit další pro pojmenování osoby – *Jmeno*, *Prijmeni*.

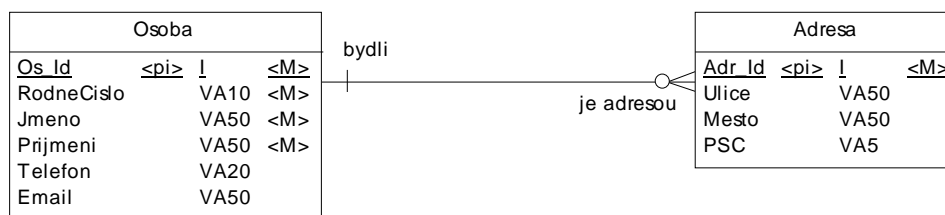
Osoba			
<u>Os_Id</u>	<pi>	l	<M>
RodneCislo		VA10	<M>
Jmeno		VA50	<M>
Prijmeni		VA50	<M>

Obr. 3.1

Ve skutečnosti lze u entity sledovat mnohem více atributů. Záleží na požadavcích náležitostí podnikového informačního systému a na tom, které z nich do modelu zakomponovat. Za mnohé lze jmenovat telefonní spojení, adresu elektronické pošty, číslo bankovního účtu aj. Jako další atribut se nabízí bydliště. V případě, že by záznam v poli bydliště obsahoval celou fyzickou adresu pobytu osoby, byla by tabulka v rozporu s první normální formou¹. Proto nelze atribut s adresou zařadit mezi atributy. Jako řešení se nabízí rozdělení jednoho atributu bydliště na dílčí atributy ulice, město a PSČ. Toto řešení lze považovat za vhodné do té doby, než bude nutné u osoby evidovat více adres. Tuto situaci lze v datovém modelu vyjádřit pomocí vztahů k jiným entitám. Namísto atributu bydliště bude datový model obsahovat entitu *Adresa*, s níž bude entita *Osoba* ve vztahu 1:N tak, jak je

¹ Normální formy představují stupně normalizace databáze. Více informací o normalizaci lze nalézt v [9]

uvedeno na obrázku 3.2. Stejným způsobem lze postupovat i v případě nutnosti evidovat více elektronických adres, nebo telefonních čísel.

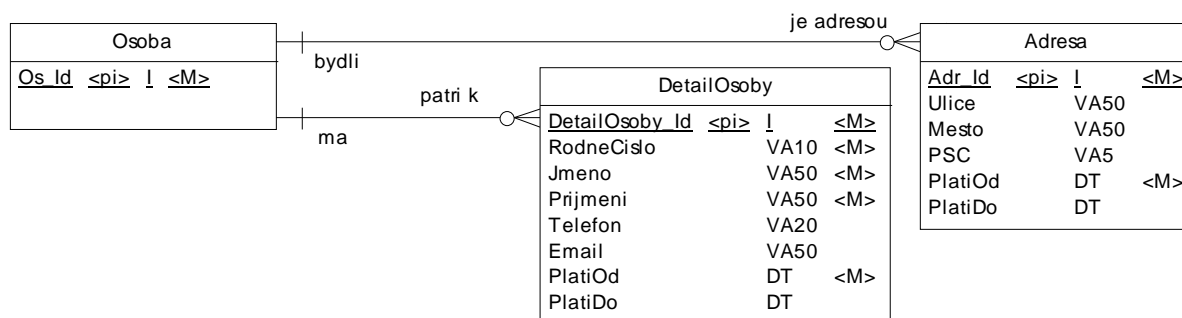


Obr. 3.2

Datový model na obrázku 3.2 dobře poslouží k vytvoření databáze lidí a kontaktů na ně. Bude-li však podnik chtít evidovat i historické informace o svých zaměstnancích, je nutné tento model modifikovat. V případě entity *Adresa* lze tento požadavek splnit přidáním atributů, které vymezují platnost vybrané instance. Při změně bydliště zaměstnance se tak původní adresa neodstraní, ale pouze se ukončí její platnost. K tomuto účelu použijeme atributy *PlatiOd* a *PlatiDo*.

Vyjdeme-li ze situace, která je znázorněna na obrázku 3.2, nelze tento jednoduchý postup aplikovat v případě změny některého z atributů instance entity *Osoba*. V tabulce *Adresa* by se tak mohl vyskytnout záznam, který bude odkazovat na již neplatnou položku z tabulky *Osoba*. Novému záznamu v tabulce *Osoba* by pak bylo nutné přidělit adresu novou.

Řešení takové situace je znázorněno na obr. 3.3. Z entity *Osoba* odebereme všechny neklíčové atributy a použijeme je v nové entitě, kterou nazveme *DetailOsoby*. Tu navíc doplníme o atributy, které vymezují validitu záznamu, jako v případě entity *Adresa*.



Obr. 3.3

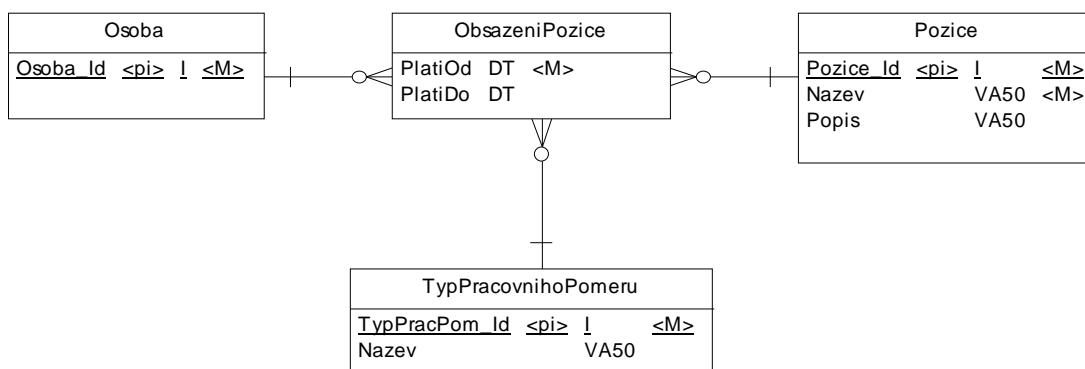
[1] [4] [5]

3.2. Zaměstnanci

Předchozí podkapitola by měla posloužit jako vzor při vytváření datového modelu pro evidenci osob a dalších informací s nimi spojených. Následující řádky jsou zaměřeny na vztahy uvnitř podniku. Jedná se především o vztahy mezi podnikem samotným a lidmi uvnitř podniku.

V předchozí kapitole je nahlíženo na entitu *Osoba* jako na zaměstnance. V praxi existuje mnoho podniků, které evidují záznamy i o tzv. nezaměstnancích. Z tohoto důvodu nese entita obecnější označení *Osoba*, jež generalizuje obě skupiny.

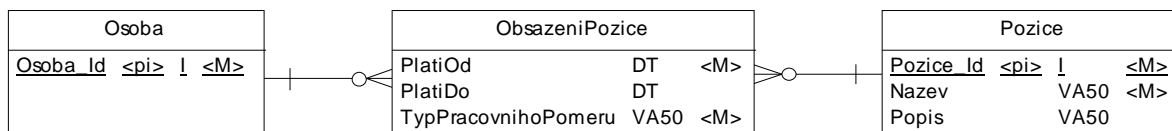
Eviduje-li podnik ve svém IS i jiné osoby než zaměstnance, je nutné je od sebe náležitě odlišit. Pomocí datového modelu lze několika způsoby z osoby „udělat“ zaměstnance. Osoba je obvykle při nástupu do nového zaměstnání dosazována na některou z pracovních pozic. Interpretace tohoto procesu v datovém modelu pak může vypadat jako na obrázku 3.4. V tomto modelu jsou znázorněny tři nové entity. Entita *Pozice* reprezentuje tabulku se seznamem pozic, které v podniku existují (např. náměstek ředitele, sekretářka útvaru, ...). Podobnou funkci v datovém modelu plní entita *TypPracovníhoPomeru*, s tím rozdílem, že bude obsahovat seznam druhů pracovního poměru (např. hlavní pracovní činnost, dohoda o provedení práce, ...). Provázání osoby s vybranou pracovní pozicí a typem pracovního poměru je realizováno pomocí vztahů těchto entit s entitou *ObsazeniPozice*. Atributy definující platnost záznamu entity *ObsazeniPozice* umožňují mimo jiné evidovat kariérní postup zaměstnanců a uchovávat historii obsazovaných pozic. Při změně pracovní pozice zaměstnance se při vytvoření nové instance *ObsazeniPozice* původnímu záznamu ukončí platnost.



Obr 3.4

V modelu na obrázku 3.5 je uvedena modifikace předchozího modelu. Namísto entity *TypPracovníhoPomeru* byla entita *ObsazeniPozice* doplněna o stejnojmenný atribut.

Hodnoty, které byly původně obsahem tabulky *TypPracovnihoPomeru* jsou doplněny do seznamu povolených hodnot nově vytvořeného atributu. Tento způsob lze použít v případě, že by tabulka se seznamem typů (v tomto případě *TypPracovnihoPomeru*) obsahovala krátký seznam záznamů, který by se jen zřídka měnil. Jelikož tomuto kritériu potenciální záznamy v tabulce typů pracovních poměrů vyhovují, lze tento přístup v tomto případě uplatnit. Nutno však připomenout, že toto řešení snižuje celkovou variabilitu budoucí databáze.



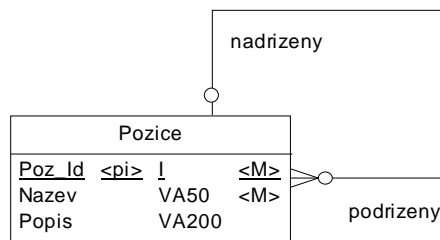
Obr 3.5

[1] [4] [5] [6] [12]

3.3. Organizace podniku

Nejčastějším vztahem mezi lidmi uvnitř podniku je vztah nadřízený - podřízený. Tento vztah vychází především z organizační struktury definované vnitřními směrnicemi každého podniku. Organizační struktura je pevně spojena s podnikatelskou činností podniku.

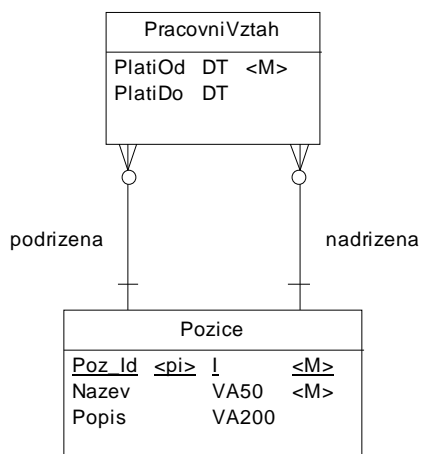
Jednoduchou organizační strukturu lze vyjádřit modelem z obrázku 3.6. Entita z předchozí kapitoly je doplněna o typ vztahu, kterému říkáme rekurzivní. Při tomto typu relace je datová entita spojena sama se sebou. Tento přístup umožňuje dynamické vytváření vztahů mezi instancemi entity. Vztah je oboustranně volitelný, jelikož pozice na nejvyšší úrovni organizační struktury již nemá nikoho, s kým by mohla navázat vztah podřízenosti. Počet podřízených (nadřízených) je v modelu znázorněn pomocí kardinality 1:N. Tento model odpovídá případu z reality, kdy má každý podřízený pouze jednoho přímého nadřízeného a může být zároveň nadřízeným několika dalších podřízených.



Obr 3.6

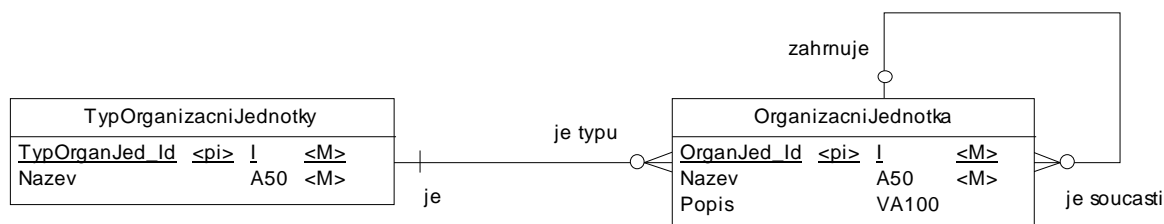
Uvedený model lze nalézt v modelech databází podnikových informačních systémů těch podniků, které využívají jednoduchou stromovou organizační strukturu. Přejít podniku na jiný typ organizační struktury ovšem tento model neumožňuje. Podnikový informační systém by měl v první řadě být pro podnik přínosem, nikoli limitujícím faktorem. Cílem by proto měl být takový model organizační struktury, který by takový přechod na jiný typ umožňoval a zároveň věrně zachycoval současný stav. Model z obrázku 3.6 nelze použít ani v případě, kdy by podnik vyžadoval evidenci historie změn v organizační struktuře.

Řešení pro oba výše zmíněné nedostatky nabízí model na obrázku 3.7. V tomto případě je použito nepřímé rekurze. Datová entita *Pozice* je spojena sama se sebou prostřednictvím entity *PracovniVztah*. To umožňuje přidělit vybrané pozici neomezené množství nadřízených i podřízených pozic. Současně lze rovněž evidovat historii organizační struktury. K tomuto účelu opět poslouží atributy vymezující platnost záznamu *PlatiOd* a *PlatiDo*.



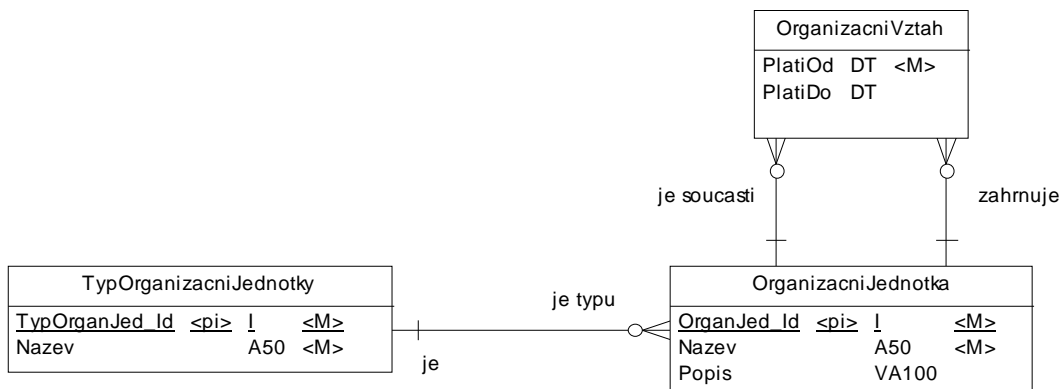
Obr. 3.7

Větší podniky využívají kromě výše zmíněné struktury i strukturu organizačních jednotek podniku. Mezi tyto jednotky patří pobočky, divize, útvary atd. Struktura některých podniků obsahuje takovýchto jednotek více. Na druhou stranu jsou i takové podniky, kde je počet organizačních jednotek minimální. Dobrý datový model by tedy měl umožnit zachytit oba případy. Model na obrázku 3.8 nápadně připomíná původní datový model organizační struktury. Model je doplněn o entitu *TypOrganizacniJednotky*, která představuje typy organizačních jednotek podniku. Tedy např. pobočky, divize, atd. Vztah k entitě *OrganizacniJednotka* je jednostranně povinný typu 1:N. Každá organizační jednotka tak musí být jednoho typu. Rekurní vztah entity *OrganizacniJednotka* pak představuje hierarchickou strukturu.



Obr. 3.8

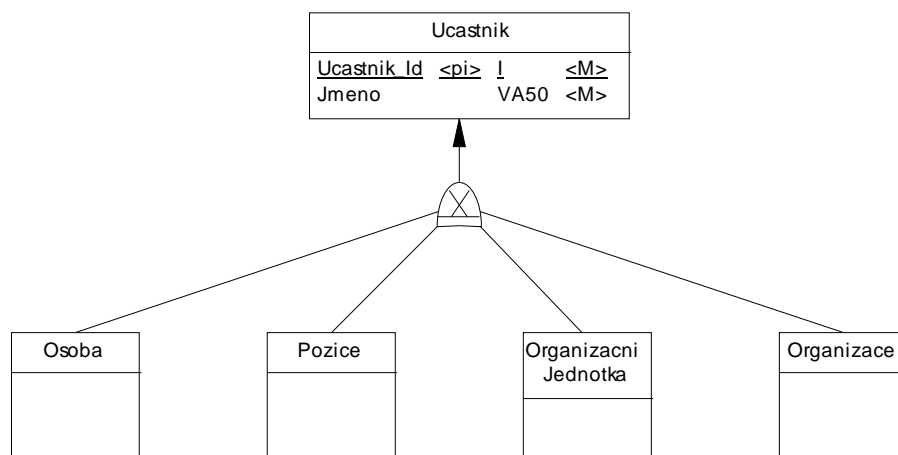
Stejně jako v případě entity *Pozice* a jejího rekurzivního vztahu (obr. 3.6), trpí model z obrázku 3.9 stejnými nedostatky. Není natolik univerzální, aby umožňoval změnu typu struktury organizačních jednotek. Tento nedostatek lze odstranit stejným způsobem jako v případě organizační struktury. Výsledek zobrazuje model na obrázku 3.9. Nepřímá rekurze opět nahradila původní přímou, tentokrát však přes entitu *OrganizacniVztah*. Možnost sledovat změny organizační struktury v čase je i zde získána pomocí atributů vymezujících platnost záznamů.



Obr. 3.9

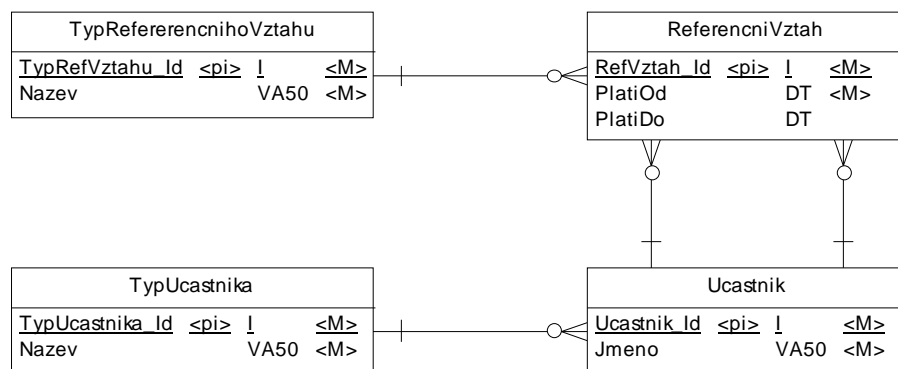
Jak vyplývá z výše uvedených diagramů, jednotlivé organizační jednotky se chovají podobně jako pozice v organizační struktuře. V některých podnicích jsou oba tyto prvky součástí jedné organizační hierarchie.

Pokud bychom byli schopni nalézt společný pojem pro zmíněné entity, lze model převést do obecnější roviny. Pro telefonního operátora není z podstaty věci důležité, zda je jeho zákazníkem fyzická osoba, nebo nadnárodní společnost. Každý z nich představuje výskyt společné entity, která zastupuje zákazníky onoho operátora. Pro tyto účely je možné zavést pojem „účastník“. Obrázek 3.10 znázorňuje generalizaci osob, pozic, organizačních jednotek a samotných organizací pod entitu *Ucastnik*.



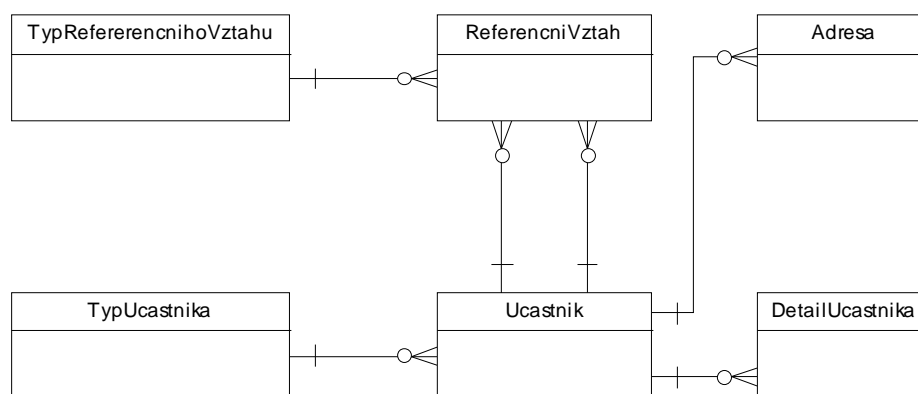
Obr. 3.10

Kromě entity *Ucastnik* je nutné pojmenovat další entitu. Ta by měla vyjadřovat vztahy mezi jednotlivými účastníky, tzn. vztahy mezi osobami a pozicemi, mezi pozicemi navzájem, organizačními jednotkami a organizacemi, atd. Pro tuto entitu je v modelu 3.11 použito označení *ReferencniVztah*. Použitelné vztahy mezi účastníky vycházejí z entity *TypReferencnihoVztahu*, která je s entitou ve vztahu 1:N. Vztah je k entitě *TypReferencnihoStavu* povinný. Každý referenční vztah musí být nějakého typu. Stejně tak každý účastník musí být nějakého typu prostřednictvím relace na entitu *TypUcastnika*.



Obr. 3.11

Na závěr této části je na obrázku 3.12 uveden diagram složený z jednotlivých modelů kapitoly. Kromě zobecnění vztahů mezi účastníky je do diagramu zakomponováno sledování změn týkajících se účastníka jako v případě osoby na diagramu 3.3.



Obr. 3.12

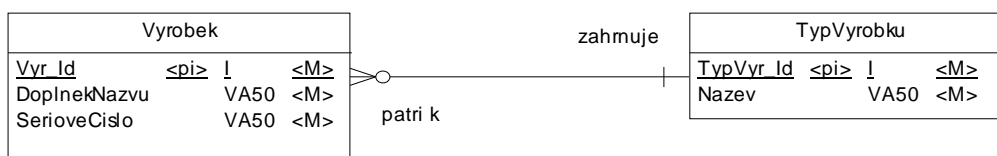
[1] [4]

4. Objekty v podniku

Tato kapitola je zaměřena na předměty, se kterými lidé z podniku přicházejí do styku. Takovým nejdůležitějším předmětem je pro výrobní podnik samotný výrobek. Výrobní program podniků obvykle zahrnuje několik typů výrobků.

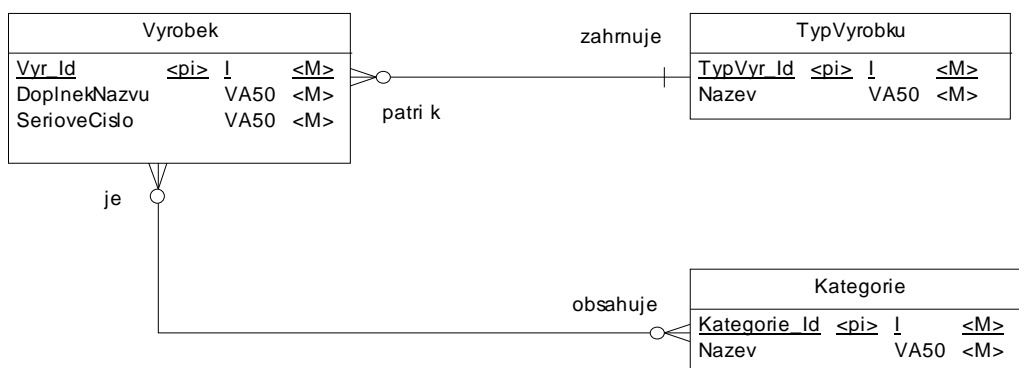
4.1. Rozdělení výrobků

BMW řady 7 je příkladem typu výrobku. Konkrétní výrobek pak představuje BMW 740 se sériovým číslem AX159753. Tento příklad lze aplikovat do modelu na obrázku 4.1. Vlastnosti jednoho typu výrobku vyjadřují atributy entity *TypVyrobu*. Na druhou stranu entita *Vyrobek* má atributy, které jsou specifické pro konkrétní výrobek. Počet sledovaných charakteristik entit a jejich struktura závisí na výrobním sortimentu jednotlivých podniků.



Obr. 4.1

Z předchozího modelu je patrné, že každý výrobek patří do jedné skupiny typu výrobku. V případě, že by se měly výrobky třídit ještě podle jiných kritérií, nelze výše znázorněný model použít. Zmíněný automobil BWM 740 patří svou konstrukcí mezi sedany. Rovněž ho však můžeme zařadit mezi luxusní automobily. Toto škatulkování umožňuje rozšířený model na obrázku 4.2. Zmíněná kritéria (sedan, luxusní automobily) jsou výskyty entity *Kategorie*. Přiřazení těchto kritérií k vybraným výrobkům je realizováno relací typu N:M mezi entitami *Kategorie* a *Vyrobek*.

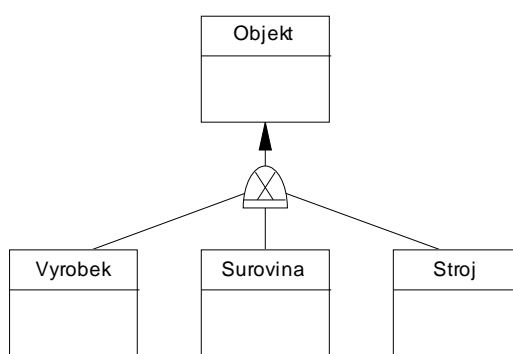


Obr. 4.2

[4] [5]

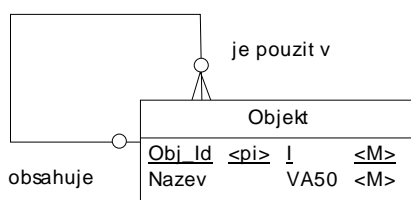
4.2. Struktura objektů

Výrobky nejsou jedinými předměty, se kterými přijdou lidé ve firmě do kontaktu. Za mnohé lze jmenovat suroviny nutné k výrobě výrobků, stroje, zařízení, inventář, atd. Tyto předměty mají z pohledu mnoha podnikových informačních systémů podobné postavení jako zmíněné výrobky. Aby bylo možné tyto předměty jednoduše zakomponovat do datových modelů, je nutné pro ně vytvořit a pojmenovat novou entitu, která by představovala obecnější pohled na tyto předměty. Takovou entitu lze označit názvem *Objekt*. Namísto entit *Vyrobek* a *TypVyrobku* použijeme novou entitu *Objekt* a *TypObjektu*. V modelu na obrázku 4.3 je naznačena generalizace specializovaných entit do obecnější entity *Objekt*.



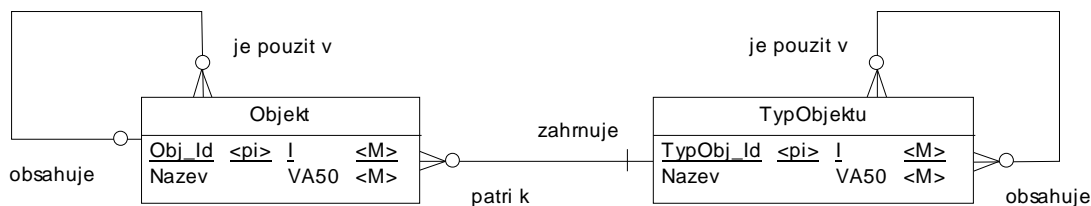
Obr. 4.3

Komplexnější výrobky, jako je například již zmiňovaný automobil BMW 740, se vyrábějí z několika dílčích výrobků. Ve většině automobilů lze nalézt motor, volant, převodovku, atd. Některé výrobky vstupují do výroby jiného výrobku jako suroviny. Pokrýt všechny objekty samostatnými entitami by bylo v případě takového výrobku (objektu) jako je automobil velmi pracné a nesystémové. Pomocí rekurze entity *Objekt* je v modelu na obrázku 4.4 znázorněno začleňování objektů do jiných instancí *Objektu*. Tyto komponenty pak mohou být zakomponovány do různých objektů. Příkladem takového dílčího objektu v automobilu je kolo, či airbag. Jelikož může být jeden objekt použit právě v jednom složitějším objektu, je kardinalita rekurzivního vztahu typu 1:N. Oboustranná volitelnost vztahu umožňuje evidovat i samostatné objekty.



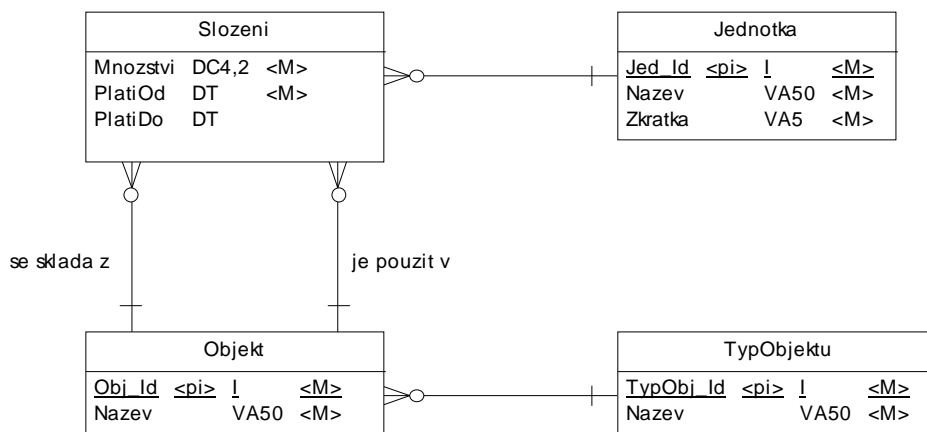
Obr. 4.4

Rekurzivní vztah z obrázku 4.4 lze rovněž použít na entitu *TypObjektu* s tím rozdílem, že je vyjádřen vztahem N:M. Na úrovni typů objektů není použití jednoho typu limitováno a lze jej využít v několika dalších typech. Tímto způsobem je vyjádřena dlouhodobá zkušenost – auta mívají 4 kola. Rekurzivní vztah entity *Objekt* pak představuje operativní stav – konkrétní auto může mít v tuto chvíli pouze 3 kola. Tuto situaci lze popsat modelem z obrázku 4.5, který kromě zmiňovaných rekurzí znázorňuje entitu *Objekt* ve vztahu s entitou *TypObjektu*.



Obr. 4.5

Začlenění některých objektů do jiných lze v modelu znázornit i jinak. Model z obrázku 4.4 navíc nelze použít pro objekty, které se vyjadřují např. v kilogramech, nebo litrech. Na obrázku 4.6 je zachycen modifikovaný model, který umožňuje práci i s nekusovými objekty. Entita *Objekt* je nepřímě spojena sama se sebou přes entitu *Slozeni*. Ta má za úkol prostřednictvím atributu *Mnozstvi* uchovávat informace o počtu vztahů mezi jednotlivými objekty. Pomocí atributů *PlatiOd* a *PlatiDo* lze v těchto vztazích sledovat realizované změny. Entita *Jednotka* pak slouží jako číselník pro jednotky veličin jako je hmotnost, délka, objem atd. Tímto způsobem lze zachytit strukturu složených objektů. Počítačová sestava se tak může skládat z 1ks procesoru, 2ks pevných disků, 4ks paměťových modulů atd. Složení sirupu proti kašli bude zcela jistě založeno na jednotkách jako jsou mililitry, nebo gramy.



Obr. 4.6

[1] [4] [11]

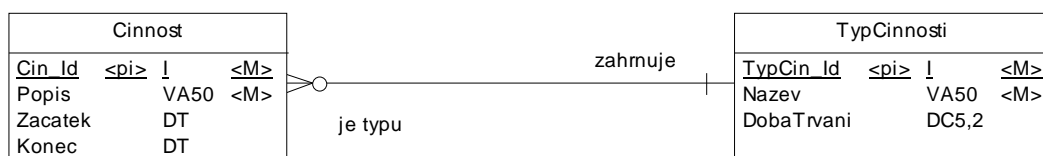
5. Činnosti v podniku

Každá organizace vyvíjí nějaké aktivity. Výrobní podniky tyto aktivity soustřeďují uvnitř podniku a jejich výstupy tvoří výrobky. Naproti tomu u organizací poskytujících služby tyto samotné činnosti představují výstupy. K zajištění přímých výstupních činností je třeba vykonat řadu podpůrných činností, například objednávku materiálu nebo pravidelnou údržbu výrobních strojů. V této kapitole budou nastíněny některé tyto činnosti z pohledu datového modelování.

5.1. Dělení činností

Podobně jako v předchozí kapitole u objektů, lze i činnosti rozlišovat na typy činnosti a samotné činnosti. Typ činnosti může představovat například instalace operačního systému na osobní počítač. Naproti tomu činností je instalace produktu MS Windows na počítač s evidenčním číslem Z159.

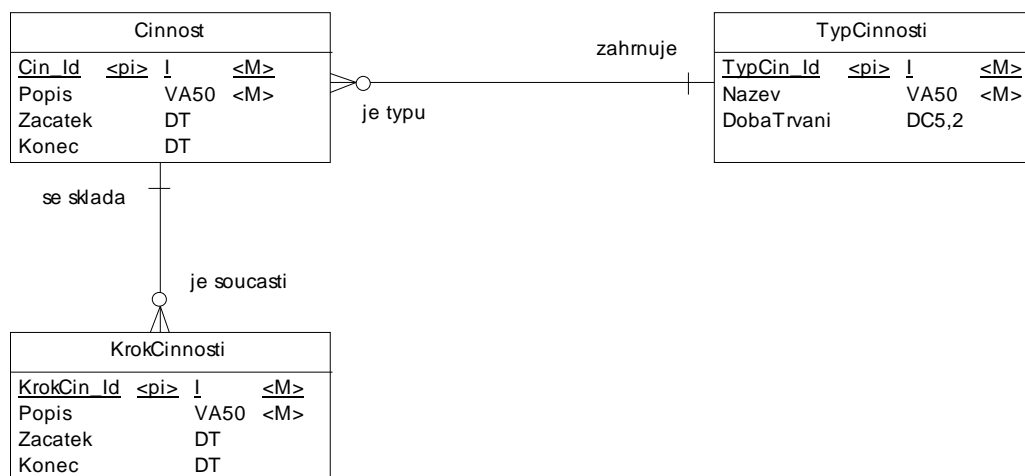
Na obrázku 5.1 je znázorněn vztah mezi entitami *Cinnost* a *TypCinnosti*. Kromě názvu a identifikátoru obsahuje entita *TypCinnosti* atribut *DobaTrvani*. Ten představuje časový údaj o typické délce trvání typu činnosti. Například instalace nového operačního systému na počítač trvá 2 hodiny. Instalace MS Windows XP na počítač s evidenčním číslem Z159 však trvala od 9:00 do 12:30. Pro tento údaj slouží entita *Cinnost*, kde se evidují konkrétní činnosti. Atributy *Zacatek* a *Konec* pak vyjadřují skutečnou dobu trvání konkrétní činnosti. *TypCinnosti* je ve vztahu s entitou *Cinnost*. Každá činnost musí mít svůj typ, proto je v modelu vztah jednostranně volitelný typu 1:N.



Obr. 5.1

V praxi bývá zvykem, že se některé složitější činnosti skládají z několika dílčích kroků. Tato skutečnost by v datovém modelu mohla vypadat jako na obrázku 5.2. Vztah mezi typem a činností zůstal zachován. Model byl rozšířen o entitu *KrokCinnosti*. Stejně jako v případě entity *Cinnost* obsahuje entita *KrokCinnosti* atributy pro časové vymezení konání jednotlivých kroků. Pomocí atributů *Zacatek* a *Konec* lze navíc vysledovat pořadí vykonaných jednotlivých kroků a jejich návaznost. Jako příklad dílčích kroků v komplexnější činnosti jako

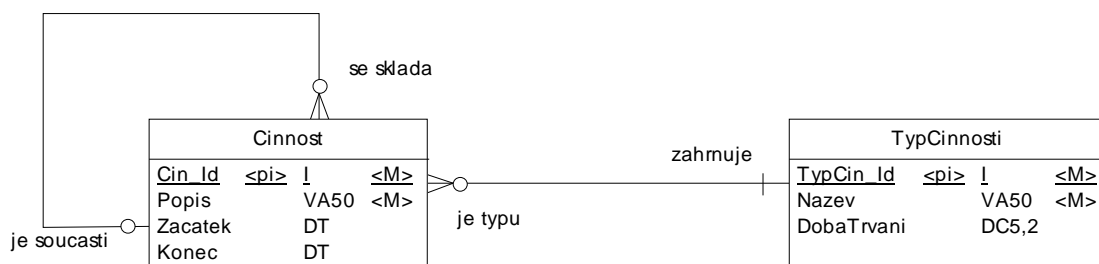
je oprava automobilu v servisu lze uvést prohlídku automobilu, počítačovou diagnostiku, samotnou opravu a testování.



Obr. 5.2

V reálných situacích se lze setkat s případy, kdy některé činnosti vstupují do jiných činností jako jednotlivé kroky. V modelu 5.2 nelze tuto skutečnost zachytit bez toho, aniž by se některé činnosti evidovaly, jako výskyty entit *Cinnost* i *KrokCinnosti* zároveň.

Řešení se nabízí v generalizaci kroků a činností do jedné obecnější entity. Jelikož můžeme nahlížet na krok v činnosti jako na samostatnou činnost, ponecháme generalizované entitě označení *Cinnost*. Přímá rekurze na entitě umožňuje využít jednu činnost jako součást jiných. Obrázek 5.3 představuje model pro zmiňovanou situaci. Rekursivní vztah je na obou stranách volitelný, protože některé činnosti nejsou součástí žádné složitější činnosti. Jelikož některé činnosti budou v sobě zahrnovat větší množství dílčích činností, zůstává ukazatel kardinality vztahu na hodnotě 1:N.



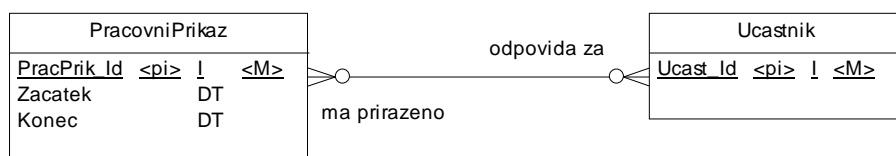
Obr. 5.3

[4] [5]

5.2. Využití lidských zdrojů v činnostech

Každá firma má svůj specifický způsob jak organizovat činnosti uvnitř podniku. Tato část je zaměřena na vykonávání činností prostřednictvím lidských zdrojů. Jinými slovy následující modely budou využívat datové entity a vztahy, které byly popisovány v předchozích kapitolách.

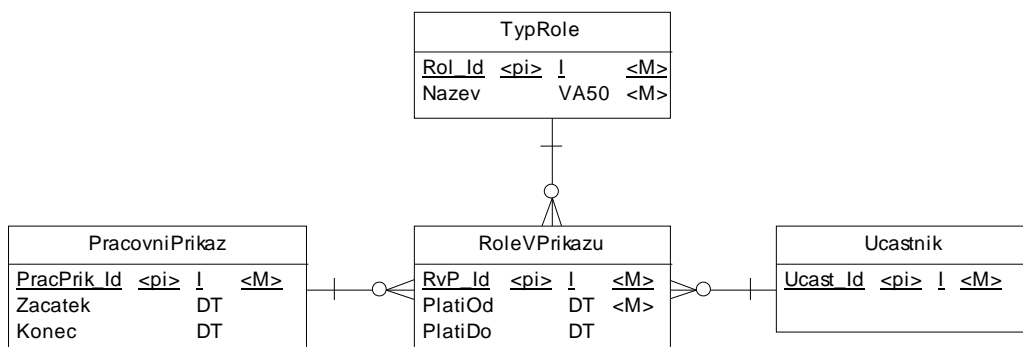
Práce lidí uvnitř firmy neprobíhá živelně, ale na základě příkazů. Ty pracovníkům ukládají, které činnosti mají vykonat. Entitu, jejíž pomocí propojíme lidské zdroje s činnostmi, nazveme *PracovniPrikaz*. V modelu 5.4 je namísto entity *Osoba* použita univerzálnější entita *Ucastnik*. Ta generalizuje jak samotné osoby (zaměstnanci, pracovníci, brigádníci,...), tak organizační jednotky a pozice. Pracovní příkaz totiž nemusí být vydán pouze osobě. Mezi entitami *PracovniPrikaz* a *Ucastnik* je oboustranně volitelná relace typu M:N. Jeden účastník může být zodpovědný za více pracovních příkazů, nebo také za žádný. Stejně tak pracovní příkaz může mít přiřazeno více účastníků a také může vzniknout bez toho aniž by byl nějaký účastník za jeho splnění odpovědný. U datové entity *PracovniPrikaz* je vhodné sledovat termíny začátku a konce plnění příkazu pomocí atributů *Zacatek* a *Konec*. Celkový počet sledovaných charakteristik u entit závisí pouze na požadavcích konkrétní organizace. Entita *Ucastnik* je v modelu 5.4 omezena pouze atributem – identifikátorem, jelikož není nutné v tomto kontextu sledovat více vlastností.



Obr. 5.4

Mezi účastníkem a pracovním příkazem mohou vznikat i jiné vztahy než pouze odpovědnost. Někdo musí pracovní příkaz vydat, další účastník by měl zkontrolovat jeho splnění, atd. Přidání všech těchto vztahů mezi entity *Ucastnik* a *PracovniPrikaz* nabízí pouze částečné řešení. A to do doby, než bude nutné za chodu systému přidat další roli účastníka v pracovním příkazu. Variabilnější řešení představuje model na obrázku 5.5. Entity *PracovniPrikaz* a *Ucastnik* jsou spojeny přes entitu *RoleVPrikazu*. Ta představuje účast rolí v pracovních příkazech. Jednotlivé role vycházejí z entity *TypRole*. Příklady typů role jsou zadavatel, programátor, asistentka, atd. Atributy *PlatiOd* a *PlatiDo* uvnitř entity *RoleVPrikazu* časově vymezují zařazení účastníky na jednotlivé role. Každá role v příkazu má přiřazeného vždy jednoho účastníka, jeden typ role a jeden pracovní příkaz. Jeden účastník, stejně jako typ

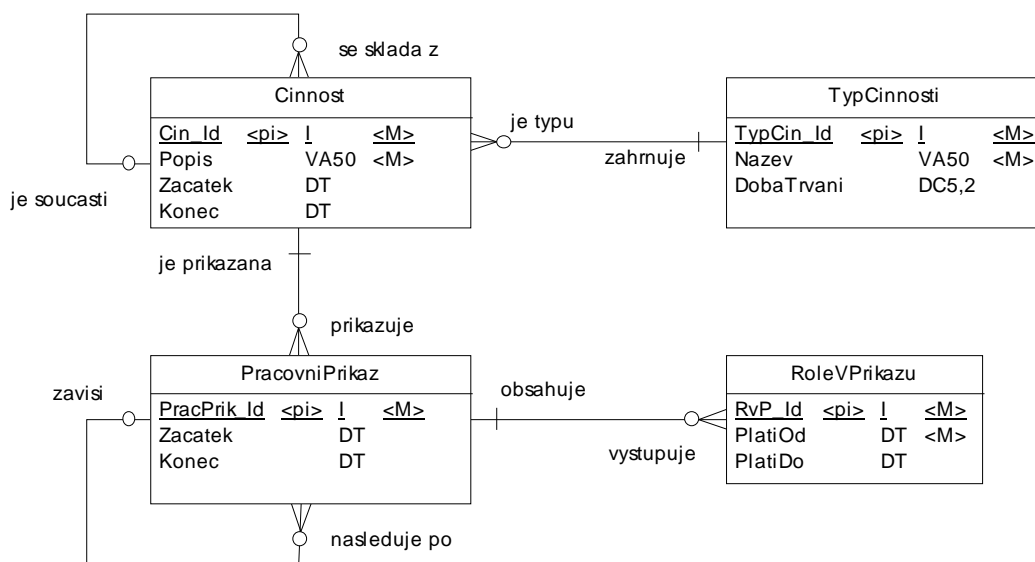
role nebo pracovní příkaz, může být přiřazen k několika rolím v příkazu. Proto mají všechny vztahy vedoucí k entitě *RoleVPrikazu* kardinalitu 1:N.



Obr. 5.5

Plnění některých pracovních příkazů je přímo závislé na splnění jiných příkazů. Programátor nemůže zahájit svou práci bez toho, aniž by byly dokončené analytické činnosti. Stejně tak mohou být vydány dlouhodobější pracovní příkazy, z nichž pak vychází detailnější příkazy operativnějšího charakteru. Pro tyto případy lze opět použít přímou rekurzi na entitě *PracovniPrikaz*, podobně jako na obrázku 5.3 v případě entity *Cinnost*.

Jak již bylo nastíněno, pracovníci se na činnostech podílejí prostřednictvím pracovních příkazů. V modelu 5.6 je doplněna druhá část tohoto vztahu. Entita *PracovniPrikaz* je s entitou *Cinnost* ve vztahu typu 1:N. Jedna činnost může být přikázána vícero příkazy. Druhá strana vztahu naznačuje, že každý pracovní příkaz přikazuje právě jednu činnost.

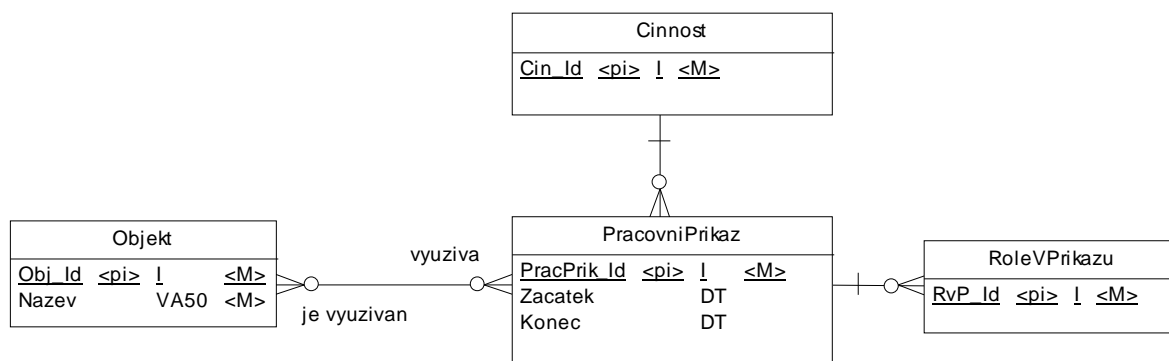


Obr. 5.6

[1] [4] [5]

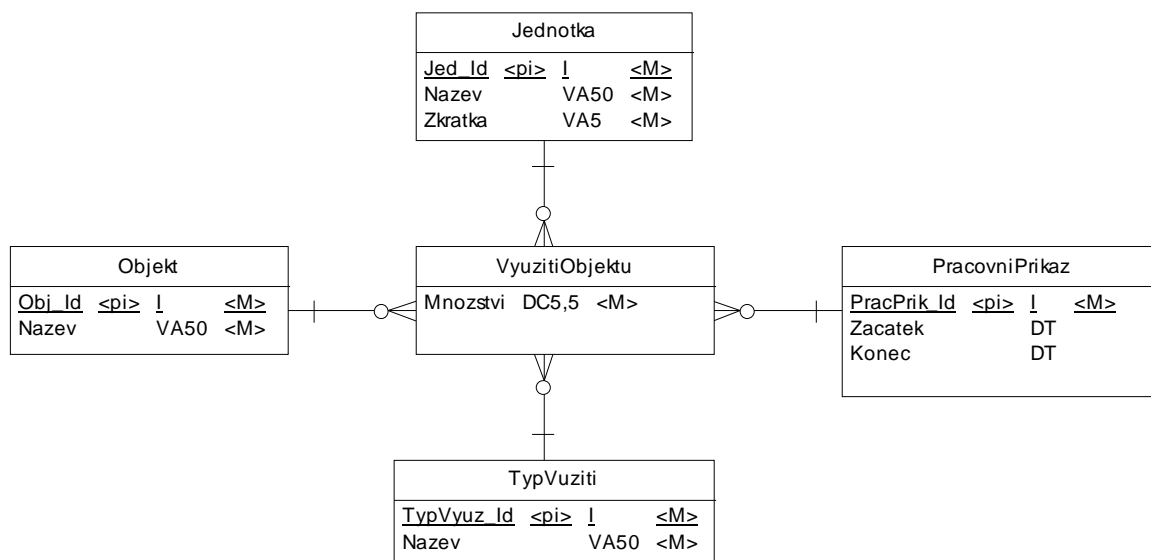
5.3. Využití objektů v činnostech

Na modelu 5.6 je znázorněno spojení činností podniku s modulem lidských zdrojů, příp. organizačních jednotek. Činnosti velmi často zahrnují i manipulaci s objekty. Za manipulaci lze považovat výrobu, údržbu, použití, spotřebu atd. Způsob zapojení objektů do činností v podniku znázorňuje model na obrázku 5.7. Aktivace objektu je opět prováděna prostřednictvím entity *PracovniPrikaz*. S entitou *Objekt* je spojena relací typu M:N. Pracovní příkaz využívá i větší množství objektů. Některé objekty pak mohou být využity několika pracovními příkazy.



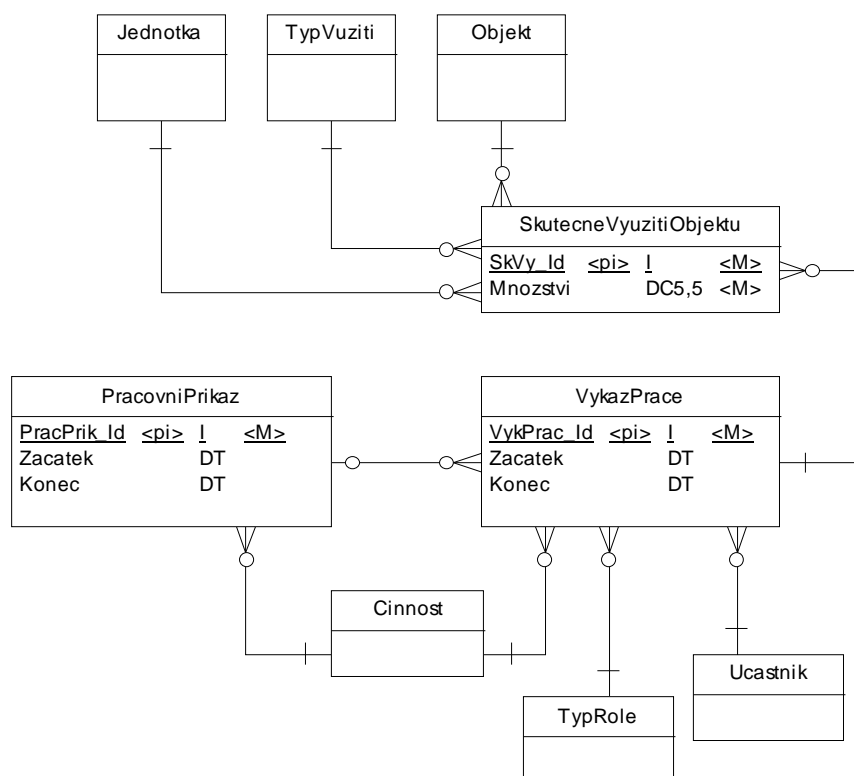
Obr. 5.7

Zapojení objektu do činností v podniku pomocí modelu 5.7 nelze však plnohodnotně použít. Objekty se obvykle vyrábějí (spotřebovávají, prodávají, ...) v nějakých jednotkách. Rovněž z modelu není zřetelné, o jaký vztah mezi entitami *Objekt* a *PracovniPrikaz* se jedná. Tento vztah je tedy nutné doplnit o entitu, která by tyto informace uchovávala. Tuto situaci lze například znázornit modelem, který je uveden na obrázku 5.8. Model byl rozšířen o vazební entitu *VyuzitiObjektu*. Ta kromě vazby mezi objektem a pracovním výkazem nese informace o počtu jednotek objektu a způsobu využití. Poslední dvě zmiňované charakteristiky jsou v modelu zastoupeny entitami *Jednotka* a *TypVyuziti*. Jednotka byla již použita ve druhé kapitole. Entita *TypVyuziti* představuje způsoby využití objektu, například výroba, oprava, spotřeba atd.



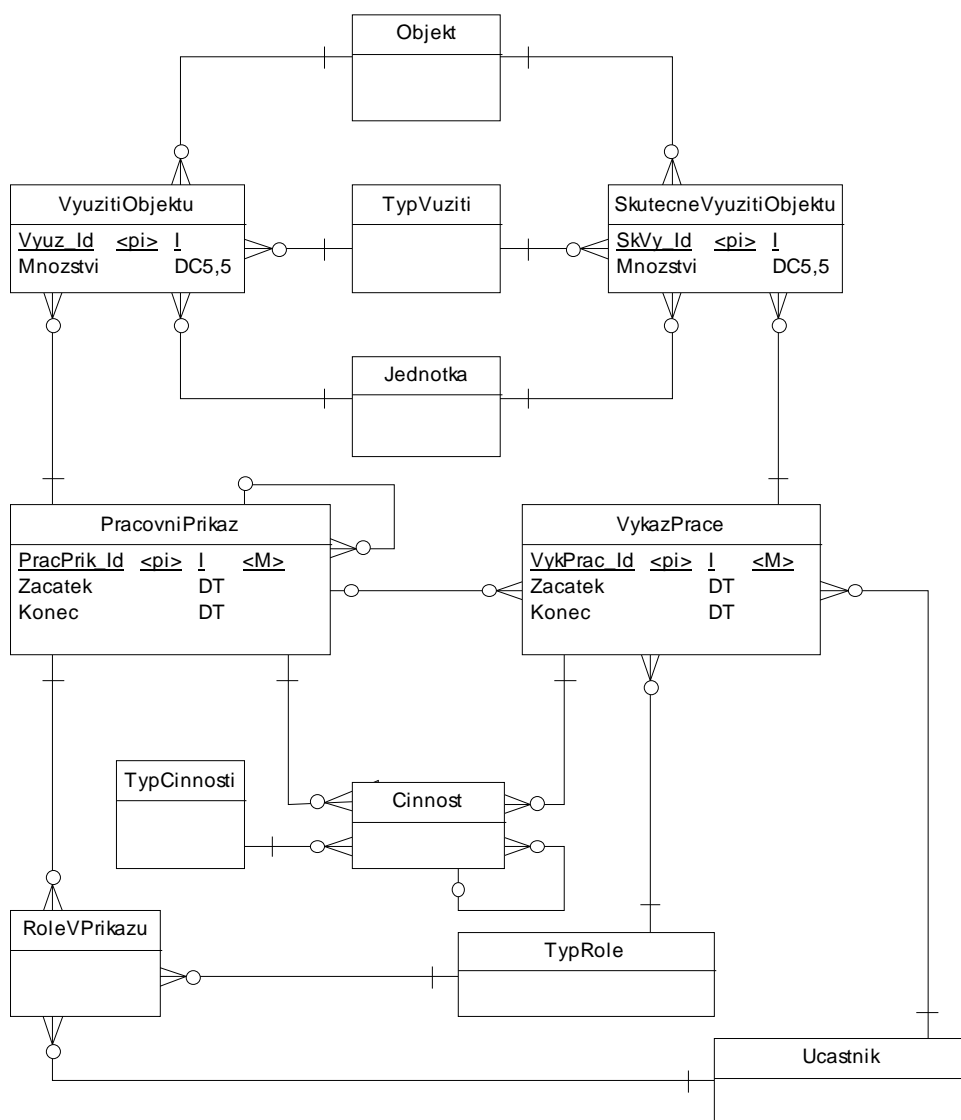
Obr. 5.8

Zatímco pracovní příkazy jsou pohledem do budoucnosti, podniky rovněž vyžadují nástroj, jehož pomocí by bylo možné sledovat plnění jednotlivých příkazů. S tím souvisí i potřeba evidovat skutečné hodnoty využitých objektů a vykonaných činností. Pro tento účel byla do modelu 5.9 zakomponována nová entita *VykazPrace*. Ta má v modelu podobný význam jako *PracovniPrikaz*, s tím rozdílem, že každý výkaz práce se vztahuje k jedné roli. S entitou *Ucastnik* je tedy *VykazPrace* spojen přímo. Atributy *Zacatek* a *Konec* představují reálnou dobu strávenou na činnosti. Reálné využití objektů v modelu zastupuje entita *SkutecneVyuzitiObjektu*. Aby bylo možné sledovat míru shody reálných výkonů s těmi plánovanými, je do modelu zaveden navíc vztah mezi entitami *VykazPrace* a *PracovniPrikaz*. Ne každý výkaz práce musí vycházet z některého pracovního příkazu. Některé pracovní činnosti vyvolané mimořádnou událostí jsou vykonávány neplánovaně. Proto je vztah oboustranně volitelný. Pro větší přehlednost jsou, až na některé výjimky, entity znázorněny bez atributů.



Obr. 5.9

Následující model 5.10 představuje celkový pohled na oblast pracovních činností uvnitř podniku. Entity a vzájemné vztahy uvnitř modelu pocházejí z dílčích modelů z této kapitoly.



Obr. 5.10

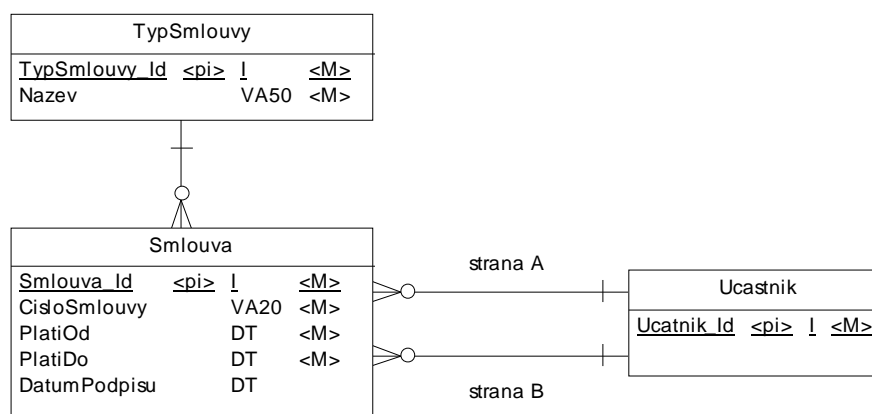
[1] [4] [5]

6. Obchodování

Předchozí kapitoly byly zaměřeny na činnosti a jejich vliv na objekty uvnitř podniku. Prodej výrobků nebo poskytování služeb je pro řadu podniků existenčním předpokladem. Na druhou stranu tyto podniky musí investovat prostředky na nákup surovin a zařízení, aby mohly své výstupy prodávat. Cílem této kapitoly je vytvoření takových datových modelů, které popisují obchodní vztahy vznikající mezi podniky.

6.1. Smlouvy

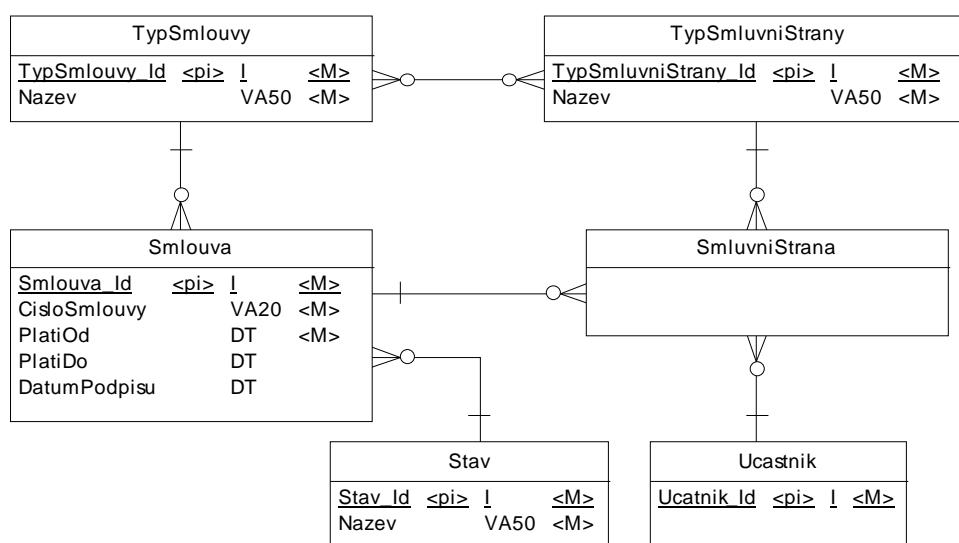
Nejčastěji vznikají obchodní vztahy na základě uzavření smlouvy. Právní předpisy definují několik druhů smluv. V praxi se nejčastěji setkáváme se smlouvami kupními, smlouvami o poskytování služeb, pracovními smlouvami atd. Některé smlouvy mají frekvenci používání mnohem nižší, např. smlouva o smlouvě budoucí. Ať se jedná o jakoukoli smlouvu, ve většině případů jde o vztah mezi dvěma subjekty. Existují však i takové smlouvy, kde je uvedeno více smluvních stran. Často se na smlouvách objevují smluvní strany prodávající – kupující, objednatel – zhotovitel, atd. Označení smluvních stran je bezprostředně závislé na typu smlouvy. V modelu 6.1 je oddělená samotná smlouva od typu smlouvy. Entita *Smlouva* obsahuje atributy, které s ní přímo souvisí. Tedy její číslo, platnost a datum uzavření. Každá smlouva musí být některého z typů, definovaných v entitě *TypSmlouvy*. Smluvní strany představují vazby 1:N mezi entitou *Ucastnik* a *Smlouva*. Entita *Ucastnik* již byla použita v předcházejících kapitolách a generalizuje jak fyzické osoby, organizace, příp. organizační jednotky.



Obr. 6.1

Jak již bylo uvedeno, některé smlouvy obsahují více smluvních stran. Příkladem dalšího účastníka ve smluvním vztahu může být zprostředkovatel nebo dopravce. Smlouvy

vycházející z modelu 6.1 mohou být realizovány pouze mezi dvěma účastníky. Aby bylo možné přiřadit jich do smlouvy více, je nutné především upravit vztah mezi entitami *Smlouva* a *Ucastnik*. Na obrázku 6.2 je znázorněno jedno z možných řešení. Do vztahu entit *Ucastnik* a *Smlouva* byla přidána nová entita - *SmluvniStrana*. Ta slouží pro evidenci účastníků, kteří vystupují jako smluvní strany ve smlouvách. Každý účastník navíc vystupuje ve smlouvě jako instance entity *TypSmluvniStrany*. Typy smluvní strany, které mohou být ve smlouvě zastoupeny, jsou definovány vztahem entit *TypSmlouvy* a *TypSmluvniStrany*. Tento vztah svým způsobem reprezentuje pravidla, jaké smluvní strany na které smlouvě mohou vystupovat. Relace mezi entitami *Stav* a *Smlouva* indikuje, v jakém stavu se smlouva nachází. Ta může být například v přípravě, čekající na podepsání, vrácena k přepracování atd.

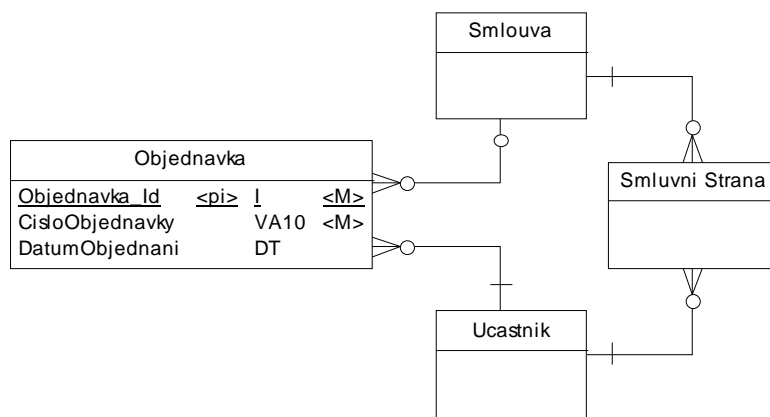


Obr. 6.2

[4]

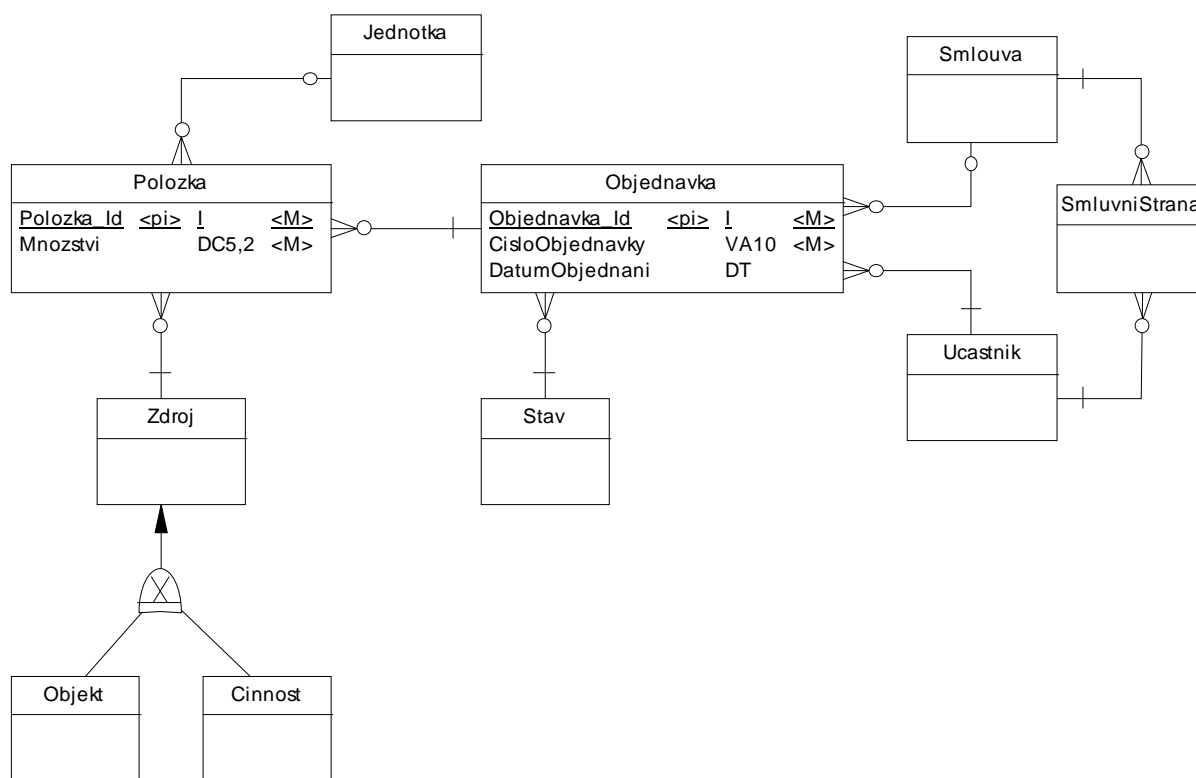
6.2. Podnikové dokumenty

Ne všechny obchody vycházejí ze smluv. Mnoho menších obchodů se realizuje pouze na základě objednávky. Objednávku lze vystavit na základě předem uzavřené smlouvy nebo nezávisle na ní. To je příklad jednorázového nákupu, který se řídí běžnými obchodními podmínkami. Obrázek 6.3 znázorňuje zmiňované způsoby vystavení objednávky. Vztahy entity *Objednavka* k entitám *Smlouva* a *Ucastnik* jsou typu 1:N. V případě vztahu k entitě *Smlouva* je znázorněn jako volitelný na obou stranách. Ve druhém vztahu je výskyt entity *Ucastnik* povinný. Relace mezi účastníkem a smlouvou vychází z předchozího modelu. Atributy entity *Objednavka* představují obvyklé náležitosti objednávky, tedy její číslo a datum uskutečnění.



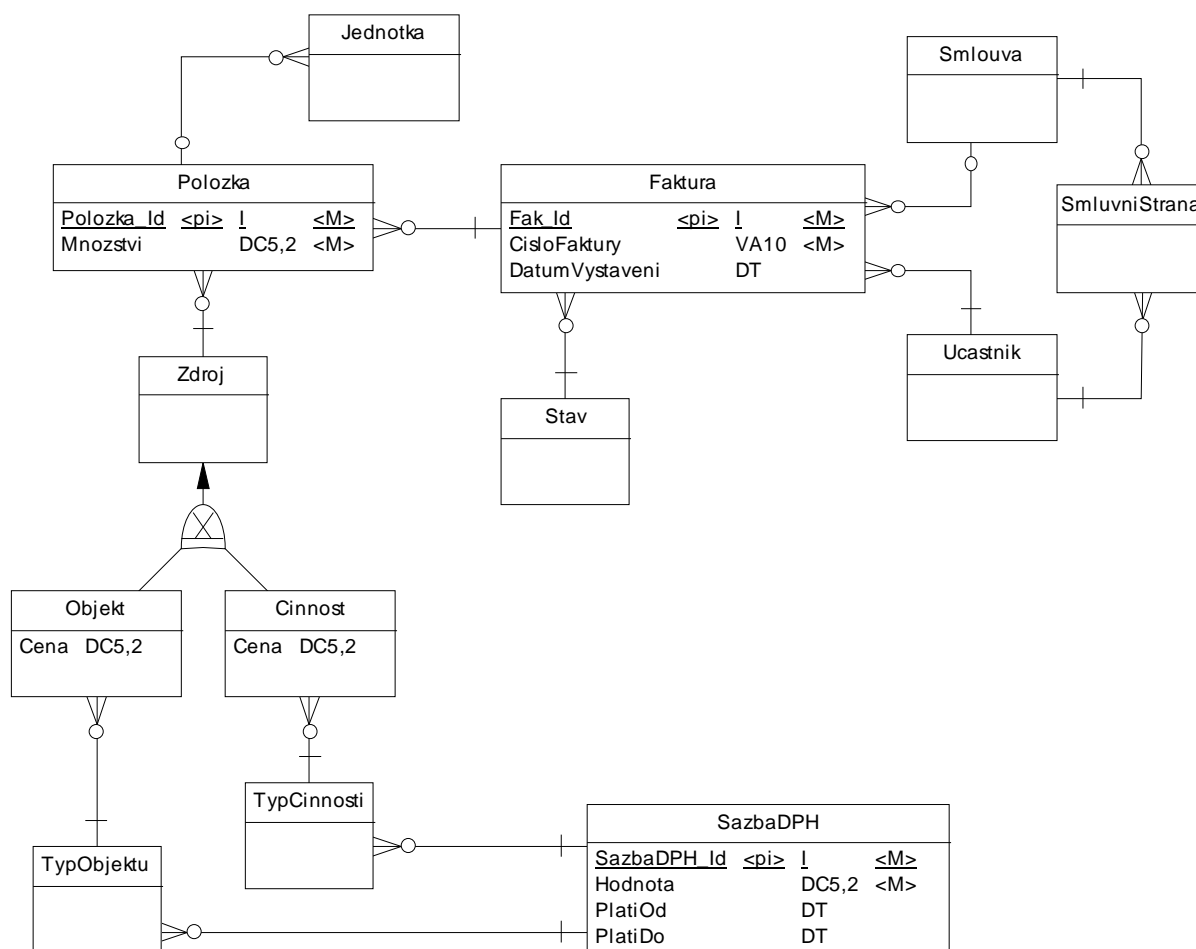
Obr. 6.3

Kromě zmíněných náležitostí objednávek nejsou v modelu uvedeny ty, kvůli kterým jsou objednávky vystavovány. A to zboží nebo služba, která je poptávána. Ty vystupují v objednávce jako její jednotlivé položky. V položce bývá kromě názvu produktu uvedeno i jeho množství a informace o tom, v jakých jednotkách je produkt objednáván. Takovouto objednávku znázorňuje model na obrázku 6.4. Pravá strana modelu zůstává nezměněna. Na druhé straně jsou k objednávce připojeny objednávané produkty. Ty do vztahu s objednávkou vstupují prostřednictvím entity *Položka*. Položkou v objednávce jsou výskyty entit *Objekt* nebo *Cinnost*, které jsou generalizovány do obecnější entity *Zdroj*. Počet objednávaných kusů v položce eviduje atribut *Mnozstvi* a zda se jedná o kusy, kilogramy, nebo jinou jednotku určuje vztah k entitě *Jednotka*. Ten je oboustranně volitelný, neboť služby se obvykle nevyjadřují v žádných jednotkách. Násobnost vztahu *Objednavka* a *Položka* je 1:N. Objednavka může tedy obsahovat více položek. Objednavka stejně jako smlouva může být v různých stavech rozpracovanosti. Tato vlastnost je v modelu zastoupena vazbou objednávky na entitu *Stav*.



Obr. 6.4

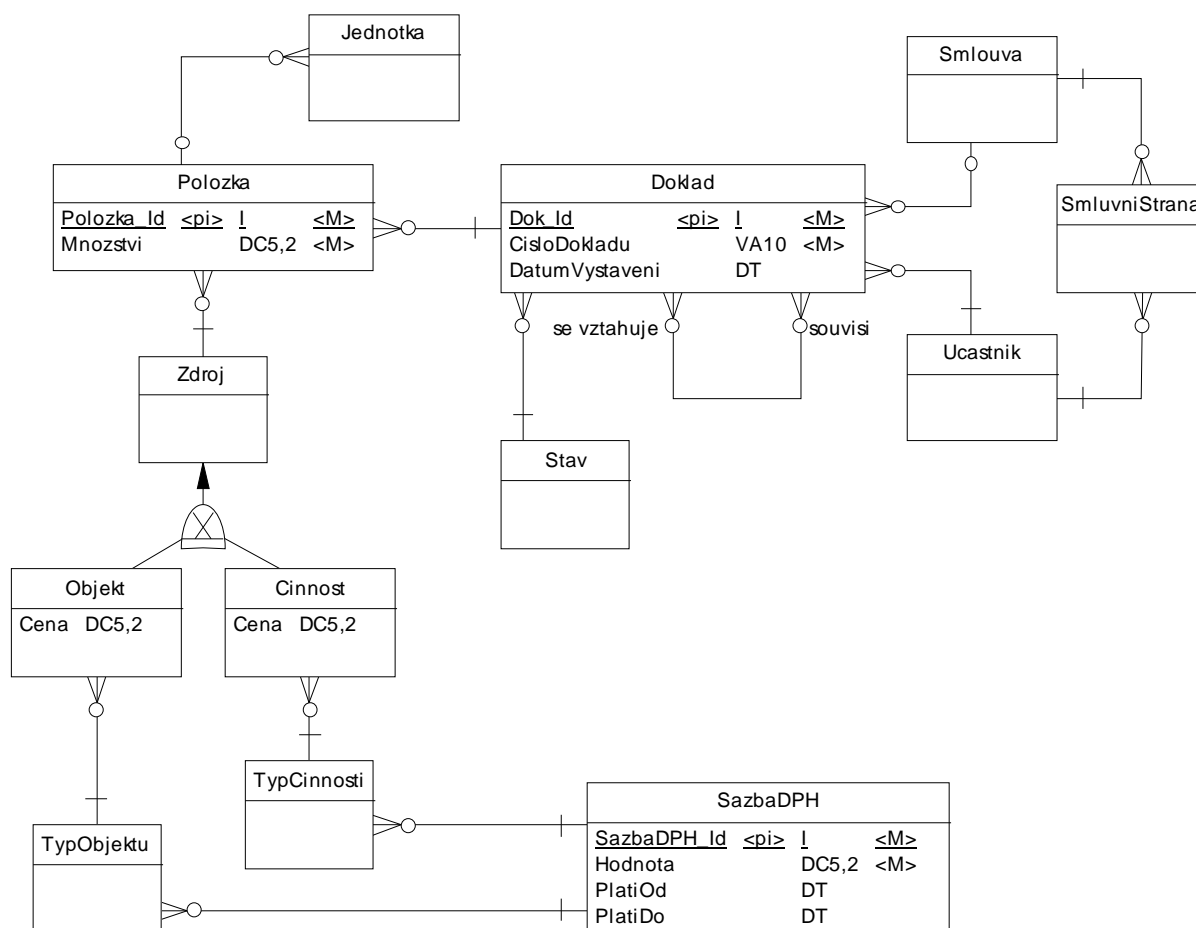
Z pohledu na model 6.4 není zřejmé, zda se jedná o objednávku vydanou nebo přijatou. Rozdíl spočívá pouze v úhlu pohledu podnikového IS na entitu *Objednavka*. Proto je tento datový model použitelný v obou případech. Tuto vlastnost má i následující model. Ten reprezentuje další krok při realizaci obchodu – fakturu. Na modelu 6.5 je patrná podobnost s modelem předchozím. Entitu *Objednavka* nahradila entita *Faktura*. Zatímco v objednávce není nutné uvádět cenu objednávaného zboží, ve faktuře je tento údaj nezbytný. Cena fakturovaného zboží (služeb) je definována atributem *Cena* při entitách *Objekt* a *Cinnost*. Každý objekt nebo činnost musí vycházet z jednoho typu. Jednotlivé typy činností a objektů jsou zatíženy státem vybíranou daní z přidané hodnoty (dále jen DPH), která se rovněž objevuje u fakturovaných položek. V současné době jsou pro různé druhy zboží a služeb používány dvě sazby DPH – 19% a 5%. Výše sazby u objektů a činností vyjadřuje vztah typu 1:N mezi entitami *TypCinnosti/TypObjektu* a *SazbaDPH*. Platnost jednotlivých sazeb vymezují atributy *PlatiOd* a *PlatiDo*. Při změně výše některé ze sazeb je tedy nutné původní sazbě ukončit platnost a vytvořit novou, platnou.



Obr. 6.5

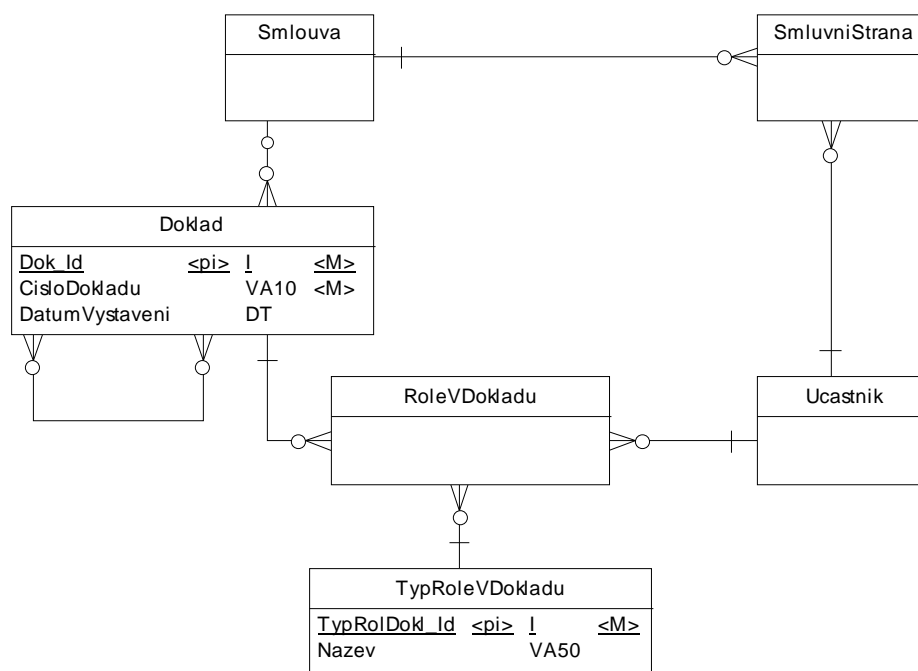
Vedle objednávky a faktury lze model použít i pro další dokumenty v podniku. Podobnou strukturu nalezneme v dodacích listech, výdejkách ze skladu atd.

V modelu 6.6 jsou podnikové dokumenty zobrazeny do entity *Doklad*. Ta je doplněna o rekurzivní vztah M:N. Tímto vztahem lze zachytit například situaci, kdy je faktura vystavena na základě přijaté objednávky.



Obr. 6.6

Způsob, kterým jsou účastníci zastoupeni ve smlouvách, lze použít i v případě ostatních dokladů. K tomuto účelu je model na obrázku 6.7 rozšířen o entity *RoleVDokladu* a *TypRoleVDokladu*, přičemž samotné zapojení účastníka do dokladu je realizováno prostřednictvím datové entity *RoleVDokladu*.



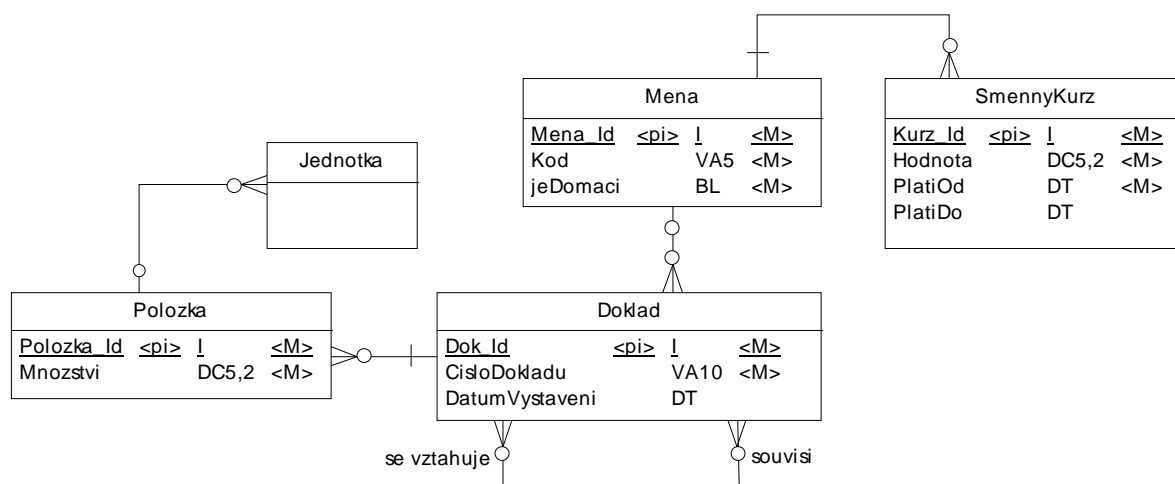
Obr. 6.7

[1] [4]

6.3. Obchody v cizí měně

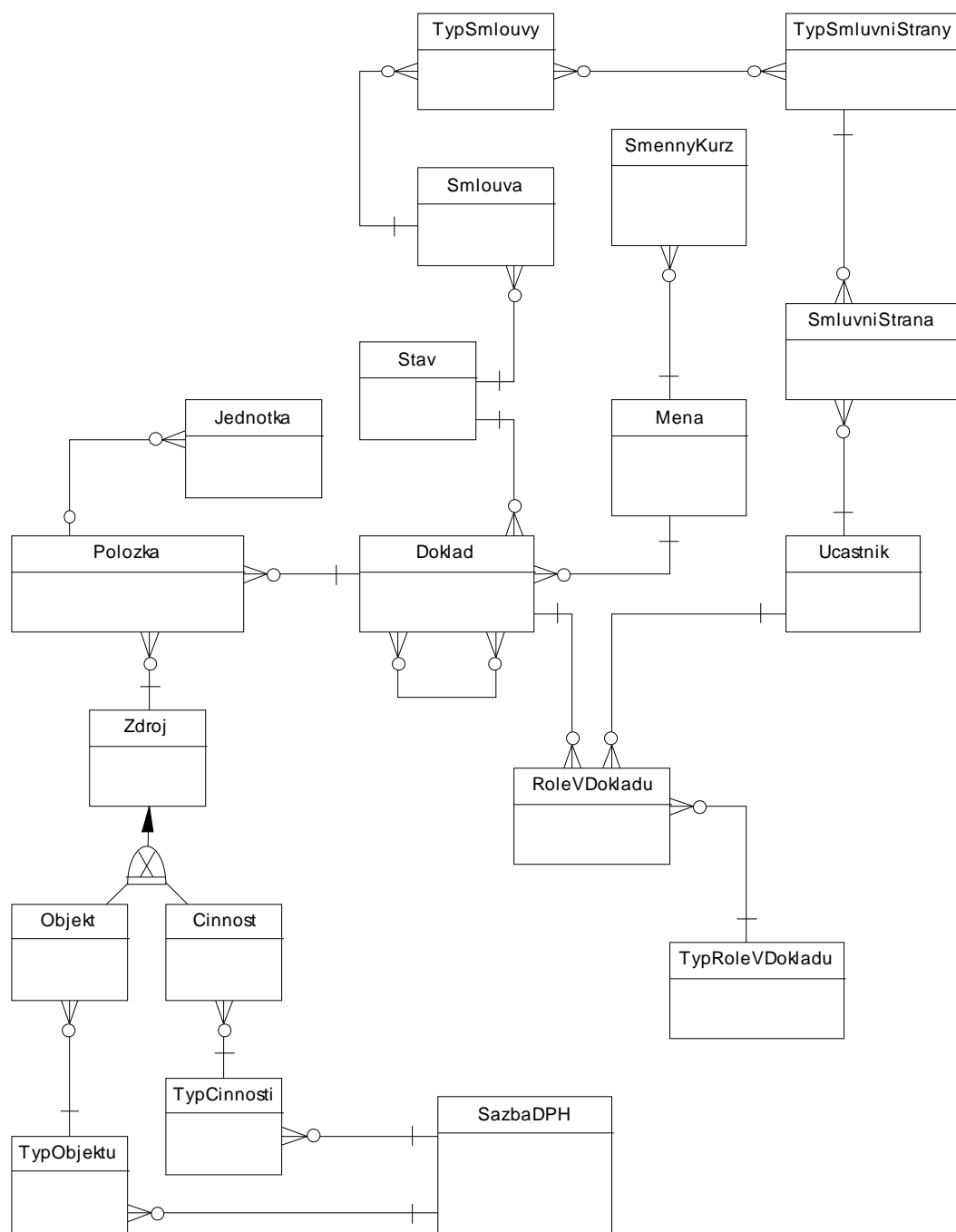
Část realizovaných obchodů tvoří obchody s firmami ze zahraničí. Platby za tyto obchody pak obvykle probíhají v některé celosvětově uznávané měně. Takovouto měnou je například euro nebo americký dolar.

Obrázek 6.8 znázorňuje model, který byl modifikován pro použití cizí měny. Volba měny probíhá na úrovni entity *Doklad* a to vztahem k entitě *Mena*. Vztah není povinný, neboť na některých dokladech se měna neuvádí. Atribut *Kod* obsahuje u jednotlivých instancí entity hodnoty jako USD, EUR, GBP, atd. Další atribut *jeDomaci* je booleovského typu. Může tedy nabývat pouze hodnot True – False. Tímto příznakem rozlišíme domácí měnu od ostatních. Aby bylo možné do faktury uvést ekvivalentní částku v jiné měně, je nutné znát převodní kurzy mezi jednotlivými měnami. Ty v modelu reprezentuje entita *SmennyKurz*. Kurz se vždy vztahuje k domácí měně, proto je mezi těmito entitami jeden vztah. V tomto případě není nutné znát kurzy mezi jednotlivými měnami. Tyto údaje by bylo možné evidovat přidáním další relace mezi entity *Mena* a *SmennyKurz*. Kurzy jednotlivých měn se mění i několikrát za den. Záleží na konkrétním subjektu jaký přístup zvolí a jak často se rozhodne kurzy obměňovat. U transakcí s vysokou hodnotou mohou chybné převody způsobit nemalé kurzové ztráty. Platnost kurzu v datovém modelu vymezují atributy *PlatiOd* a *PlatiDo*.



Obr. 6.8

Problematika spojená s obchodními vztahy podniků je velmi obsáhlá. Vše se odvíjí od aplikační oblasti a specializace jednotlivých podniků. Následující obrázek 6.9 je příkladem jednoho z přístupů k této problematice. Vznikl propojením dílčích modelů z této kapitoly.

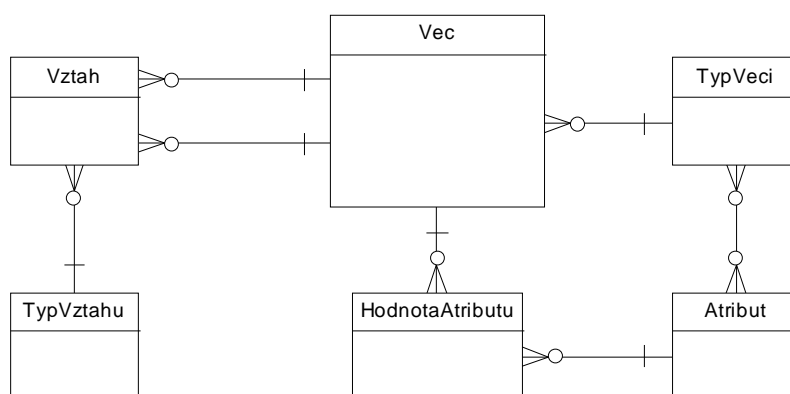


Obr. 6.9

7. Univerzální model

Následující model představuje zobecnění všech výše uvedených modelů. A nejenom ty. Pomocí diagramu na obrázku 7.1 lze popsat téměř všechno.

Svět se skládá z věcí. Proto ústředním prvkem tohoto datového modelu je entita *Vec*. Každý výskyt věci představuje instanci entity *TypVeci*. Datová entita *Atribut* zobecňuje všechny sledované vlastnosti věcí. Vztah M:N mezi entitami *TypVeci* a *Atribut* definuje, které atributy se vztahují k určitému typu věci. Například mezi atributy osoby bude s největší pravděpodobností patřit datum narození. Tento atribut však nenajde své uplatnění pro věc, jakou je například automobil. Přiřazení atributů ke konkrétním věcem zprostředkovává mimo jiné entita *HodnotaAtributu*. Použití této entity může omezit datový typ atributu. Relace mezi jednotlivými věcmi se realizují přes entitu *Vztah*. Každý vztah musí vycházet z některého typu vztahu. Uvedeným modelem lze popsat pouze binární vztahy – tedy vztahy mezi dvěma výskyty entity *Vec*. Pojmenování jednotlivých rolí v těchto relacích závisí na výskytech entity *Vec*, které do vztahů vstupují. Tato část modelu ne náhodou připomíná vztahy mezi účastníky z diagramu 3.11, které jsou založeny na stejném principu.



Obr. 7.1

Zobecnování entit a případně i celých modelů může být při vytváření reálných aplikací přínosem. Na druhou stranu přílišná míra obecnosti není v některých případech příliš žádoucí a implementovat takovýto systém obvykle přináší více problémů než užitku.

Diagram 7.1 představuje pouze další stupeň abstrakce. Uplatnění pravděpodobně nenajde v žádné aplikaci. Lze ho nicméně použít jako nejobecnější návrhový vzor pro jakoukoli databázi.

[1] [12]

Závěr

V oblasti návrhu objektově orientovaného programového vybavení nejsou návrhové vzory ničím novým. Pojem návrhový vzor se v souvislosti s návrhem databází nevyskytuje příliš často.

Snahou této práce bylo vytvoření některých datových modelů, které by odpovídaly definici návrhového vzoru. Ta je uvedena v první kapitole této práce, která je zaměřena na seznámení s problematikou návrhových vzorů. Druhá kapitola je věnována teoretickým základům datového modelování a grafickým symbolům, které jsou v následujících kapitolách použity.

Práce je primárně zaměřena na datové modely pro databázi podnikového informačního systému, nicméně výsledky mají širší použití. Databáze je rozdělena do několika částí, z nichž každou pokrývá jedna kapitola. Po teoretickém úvodu do datového modelování následuje kapitola s diagramy, které zachycují vnitřní organizaci podniku. Poznatky z této kapitoly jsou použity v kapitolách následujících. Jejich obsah je již více specializován na činnosti uvnitř podniku a jejich výstupy. Datové modely jsou postupně doplňovány a upravovány pro co nejširší použití. Závěrečná kapitola pak představuje nejobecnější pohled na datový model, s jehož pomocí lze vyjádřit množství situací z reality.

Na závěr bych rád zmínil alespoň část literatury, kterou jsem k této práci nastudoval a která mi pomohla utvořit si vlastní náhled na problematiku. Na základě této literatury a zkušeností získaných z praxe jsem v některých případech použil odlišného přístupu k řešení určitých situací. Ze studované literatury byly pro mne přínosné zejména publikace [1] a [4].

Výsledky z této práce by mohly nalézt uplatnění při řešení některých situací při navrhování databází. Přestože se v této práci jedná o návrh databáze pro informační systém podniku, případnou modifikací lze jeho použití rozšířit i na jinou aplikační oblast.

V budoucnu bych se chtěl této problematice i nadále věnovat. S využitím praktických zkušeností bych se rád zaměřil na vzory i pro jiné oblasti.

Přílohy

V této části jsou uvedeny praktické ukázky datových modelů, ve kterých jsou uplatněny poznatky z některých vzorů, uvedených v této práci.

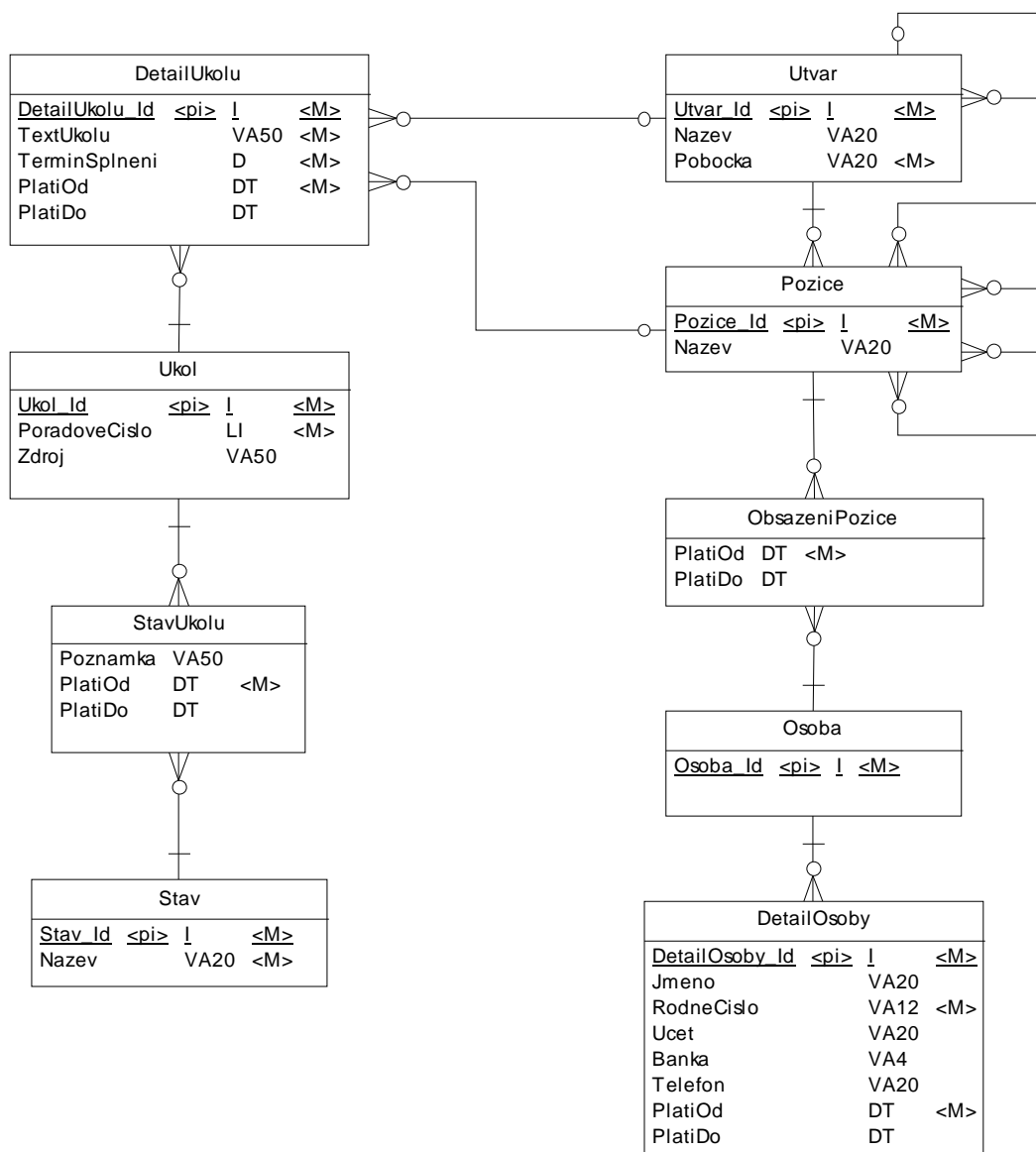
Na rozdíl od obecně pojatých předchozích modelů představují následující diagramy části podnikových aplikací, které jsou úspěšně používány v praxi. Datové modely zobrazují pouze vybranou část aplikace a ty části, se kterými přímo souvisí. Ke každému modelu je uveden komentář, co které entity a atributy představují a které vztahy mezi nimi vznikají. Z praktických důvodů jsou v diagramech znázorněny pouze ty nejdůležitější entity a atributy.

Příloha A: Kniha úkolů

Datový model na obrázku A.1 znázorňuje databázi pro modul podnikového informačního systému. Jeho prostřednictvím jsou přidělovány úkoly jednotlivým pracovním pozicím v podniku. Z tohoto důvodu je modul úzce spjat s organizační strukturou.

Tu reprezentuje pravá část modelu. Podnik má pobočky na dvou místech v ČR, v Praze a v Brně. Každá pobočka je rozdělena do útvarů. Pobočky v organizační struktuře jsou v modelu zastoupeny pouze atributem v rámci entity *Utvár*. Hierarchie útvarů v organizační struktuře je na modelu znázorněna pomocí rekurzivního vztahu entity *Utvár*. Jak je z modelu patrné, útvary jsou tvořeny pozicemi. Každý výskyt entity *Pozice* náleží k některému z útvarů. Rekurzivní relace na entitě *Pozice* modelují zástupnost jednotlivých pozic, respektive oprávněnost přiřazovat úkoly jinými pozicím. Přiřazení osob na pozice je realizováno prostřednictvím entity *ObsazeníPozice*. Doba, po kterou je osoba přiřazena ke konkrétní pozici, je vymezena atributy *PlatiOd* a *PlatiDo*. Tento přístup byl použit v některých popisovaných vzorech. Rovněž evidování historie údajů o osobě pomocí entity *DetailOsoby*. Veškeré údaje vztahující se k osobě jsou evidovány prostřednictvím entity *DetailOsoby*. Změny v osobních údajích se v databázi objeví jako nová platná instance a původnímu se ukončí platnost.

Propojení organizační struktury s úkoly je v modelu znázorněno vztahy entity *DetailUkolu* s entitami *Utvár* a *Pozice*. Těmito relacemi je znázorněno přidělování úkolů. Ty mohou být přiděleny konkrétní pozici, nebo celému útvaru. Entita *DetailUkolu* má v modelu stejný význam jako *DetailOsoby*. Jejím prostřednictvím je evidována historie jednotlivých úkolů. Informace k úkolu, které se mohou měnit (např. znění úkolu nebo termín jeho splnění), jsou uchovány prostřednictvím atributů v entitě *DetailUkolu*. Na rozdíl od těchto dat jsou atributy entity *Ukol* přiděleny pouze jednou a zůstávají neměnné. K těmto datům patří například pořadové číslo nebo zdroj, ze kterého úkol vzešel. Kardinalita vztahu mezi entitami *Ukol* a *DetailUkolu* má hodnotu 1:N. K určení platnosti i v případě entity *DetailUkolu* slouží atributy *PlatiOd* a *PlatiDo*. Vývoj úkolů je popsán pomocí entity *Stav*. Ta slouží jako číselník se seznamem všech stavů, kterých může úkol nabývat. Příkladem takového stavu je úkol v přípravě, probíhá, dokončený, atd. Přiřazení stavu k vybranému úkolu a evidence historie stavů je zprostředkováno prostřednictvím entity *StavUkolu*. K určení aktuálního stavu úkolu opět slouží atributy pro vymezení platnosti. Při přechodu do nového stavu se platnost původního stavu ukončí.



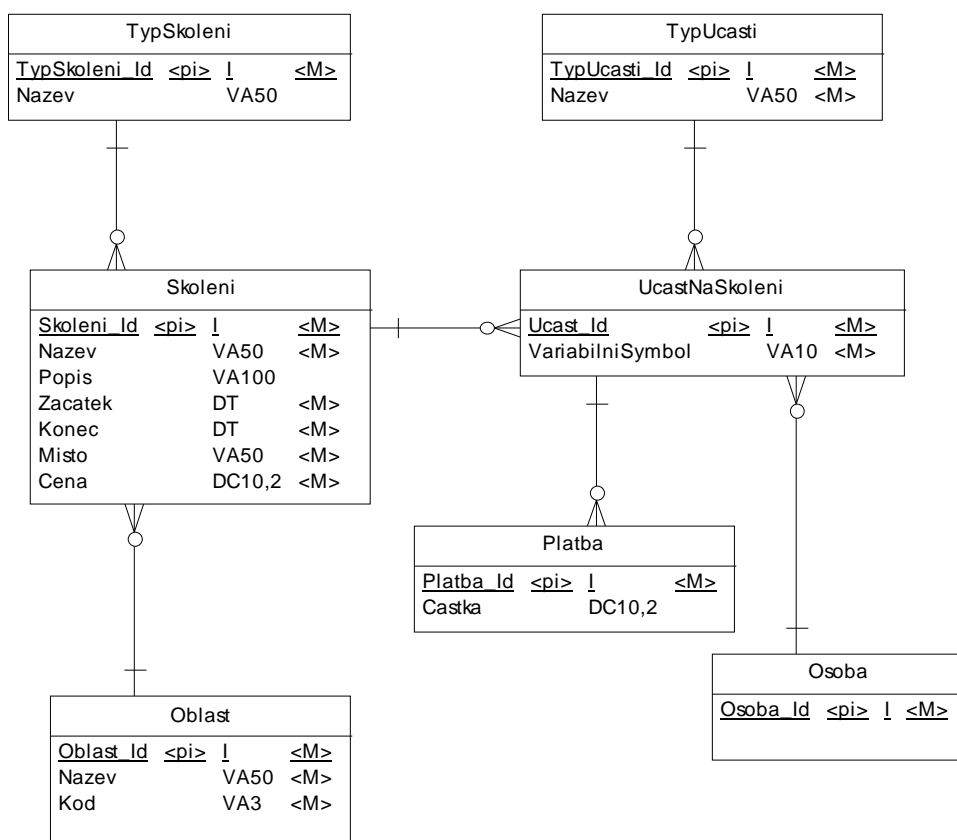
Obr. A.1

Příloha B: Školení a semináře

Další příklad použití některého z uvedených vzorů je zachycen na obrázku B.1. Na modelu je zobrazena databáze, kterou využívá aplikace pro evidenci osob na odborných školeních a seminářích.

Školení a semináře reprezentuje v modelu datová entita *Skoleni*. Ke každému školení je nezbytné uvádět od kdy a do kdy trvá, kde se koná a jaká je jeho cena. K upřesnění o jaké školení se jedná se využívá vazeb na entity *TypSkoleni* a *Oblast*. Typ vztahu 1:N znamená, že každé školení musí být nějakého typu a pro některou z oblastí.

Účast osob na jednotlivých školeních je evidována prostřednictvím entity *UcastNaSkoleni*. Většinu výskytů této entity představují posluchači. Kromě nich však na akcích vystupují další osoby například v roli přednášejících. Jednotlivé role na školeních jsou tedy generalizovány do jediné entity. Tyto role jsou diverzifikovány pomocí vazby na entitu *TypUcasti*. Tento ternární vztah je nápadně podobný vzoru z šesté kapitoly, kde byly evidovány jednotlivé role v podnikovém dokladu. Úhradu za školení, které je hrazeno účastníkem, znázorňuje entita *Platba* a její vazba na entitu *UcastNaSkoleni*.



Obr. B.1

Použitá literatura

- [1] HAY, David C. *Data Model Patterns: Conventions of Thought*. New York: Dorset House 1996, ISBN 0-932633-29-3
- [2] DVOŘÁK, Miloš. *Návrhové vzory (design patterns)*. Praha: VŠE 2003
- [3] GAMMA, E., HELM, R., JOHNSON, R., VLISSIDES, J. *Návrh programů pomocí vzorů: Stavební kameny objektově orientovaných programů*. Praha: Grada Publishing, 2003, ISBN 80-247-0302-5
- [4] ŠEŠERA, L., MIČOVSKÝ, A., ČERVEŇ, J. *Datové modelování v příkladech*. Praha: Grada Publishing, 2001, ISBN 80-247-0049-2
- [5] STEVENSON, Len. *The Data Model Resource Book – Volume 1 (A Library of Universal Data Models for All Enterprises)*. New York: Wiley Computer Publishing 2001, ISBN 0-471-38023-7
- [6] STEVENSON, Len. *The Data Model Resource Book – Volume 2 (A Library of Universal Data Models by Industry Types)*. New York: Wiley Computer Publishing 2001, ISBN 0-471-35348-5
- [7] ALEXANDER, C., ISHIKAWA, S., SILVERSTEIN, M. *A Pattern Language*. New York: Oxford University Press 1977, ISBN 0-19-501949-9
- [8] HERNANDEZ, Michael J., *Návrh databází*. Praha: Grada Publishing 2005, ISBN 80-247-0900-7
- [9] CHLAPEK, D., ŘEPA, V., *Materiály ke strukturované analýze*. Praha: VŠE 1997, ISBN 80-7079-260-4
- [10] POKORNÝ, Jaroslav. *Konstrukce databázových systémů*. Praha: Vydavatelství ČVUT 2001, ISBN 80-01-01935-7

Elektronické zdroje

- [11] Design Pattern - <http://www.lukew.com/ff/entry.asp?348>
- [12] Library of Free Data Models - http://www.databaseanswers.org/data_models

Terminologický slovník

Entita	označení obecného jasně definovaného prvku
Atribut	vlastnost entity, nebo sloupec databázové tabulky
Generalizace	způsob tvoření obecných pojmů z pojmů méně obecných nebo jedinečných
ERD	(Entity Relationship Diagram), diagram entit a vztahů