

9 Strukturovaná analýza

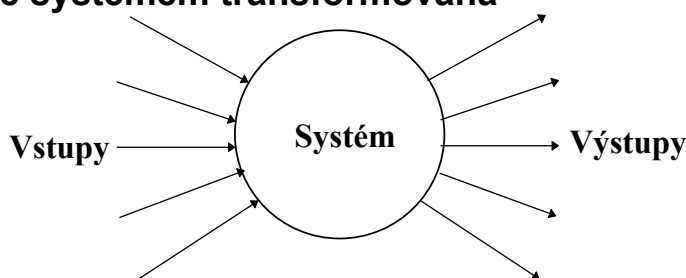
9.1 Modelovací techniky strukturované analýzy

- systém chápán jako kolekce funkcí (procesů) operujících nad daty
- **funkční (procesní) modelování**
 - základní model strukturované analýzy
 - ukazuje funkce systému, toky dat mezi systémem a okolím a mezi funkcemi, data ukládaná v systému
 - diagram datových toků (Data Flow Diagram - DFD)
- **minispecifikace**
 - popis funkcí (procesů) – co dělají
- **datové modelování**
 - ukazuje entity aplikační domény zpracovávané systémem a statické vztahy mezi nimi (typicky perzistentní data ukládaná v databázi)
 - důležitý model datově intenzivních aplikací
 - zásadní význam pro návrh databáze
 - diagram entit a vztahů (Entity Relationship Diagram - ERD)

- **datový slovník**
 - obsahuje specifikace prvků modelů
 - notace pro specifikaci informačního obsahu prvků DFD a ERD
- **modelování dynamického chování**
 - stavový diagram - stavy systému, události, akce

9.2 Funkční modelování – diagramy datových toků (Data Flow Diagram – DFD)

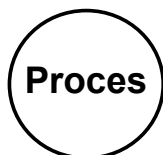
- grafická technika, modelující toky dat a jejich transformace
- informace je systémem transformována



prvky DFD:



- aktér (terminátor, externí entita) – zdroj/příjemce informace pro/ze systému,



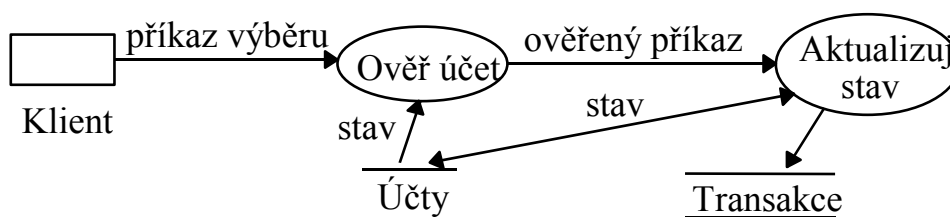
- transformuje data,



- pojmenovaný tok dat,

Datová paměť - obsahuje uloženou informaci, využitou systémem

Př)



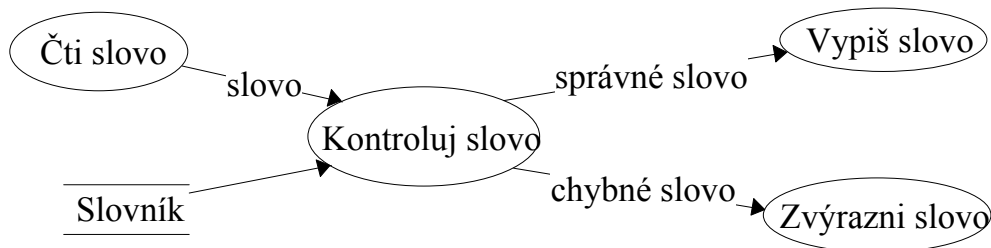
• proces

- provádí transformaci dat (logickou nebo fyzickou), jméno by mělo vyjadřovat podstatu transformace, ne generická jména
- každý proces je buď specifikován (minispecifikace) nebo reprezentován jiným DFD (víceúrovňové DFD)

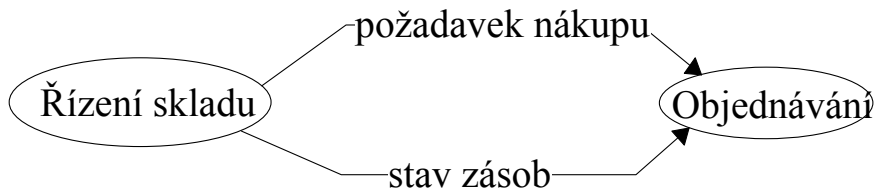
• datový tok

- reprezentace přechodné hodnoty v průběhu zpracování, pohyb dat (paketů informace), někdy také znázornění toku fyzických materiálů

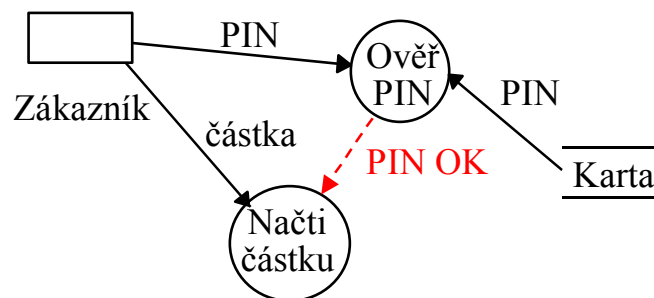
➤ **stejná data mohou mít různý význam (logická tranformace)**



➤ **datový tok nese vždy jen jeden typ paketu informace**



➤ **co není datový tok – DFD neukazuje řízení, existují rozšíření**



- **aktér**

- zdroj/příjemce hodnoty ležící vně systému, reprezentují rozhraní mezi systémem a vnějším světem

- **datová paměť**

- pasivní objekt pro uložení dat pro pozdější zpracování, modeluje data v klidu (vztah k ERD)

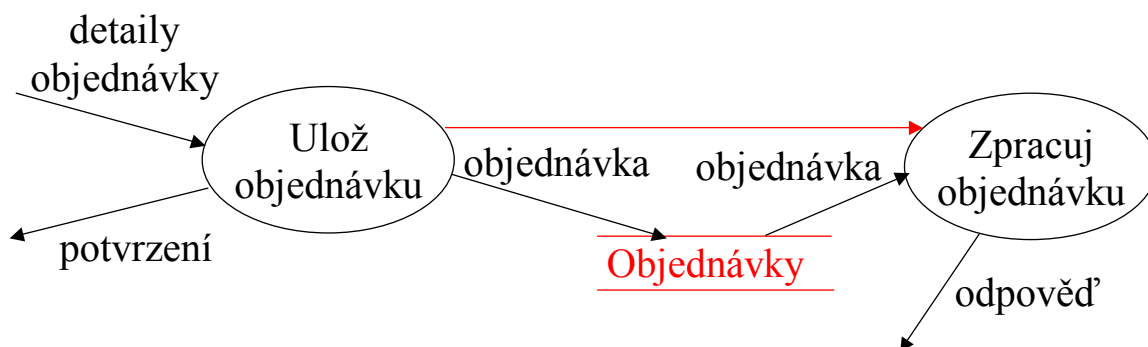
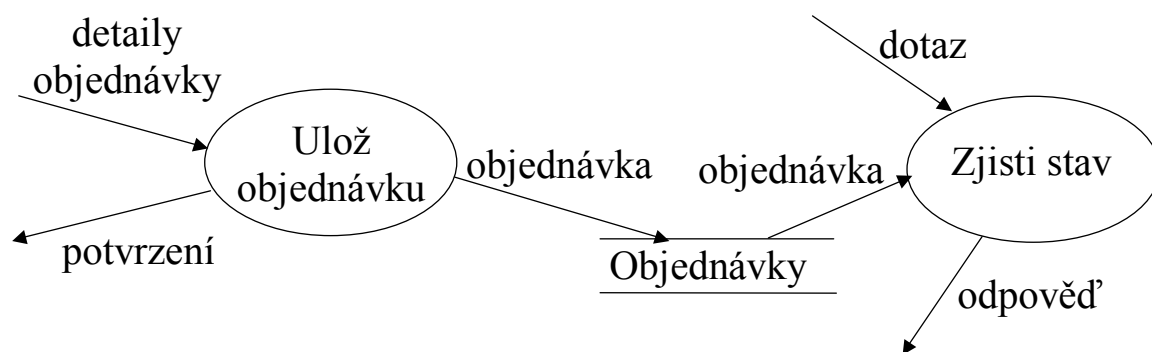
- implementace – soubor, databáze, archiv, ...

- jméno – zpravidla množné číslo od uložených paketů informace (entit)

- různé možné interpretace toku do paměti – jeden nebo více nových/zrušených/modifikovaných paketů

- různé možné interpretace toku z paměti – celý paket, více paketů, část paketu, části několika paketů

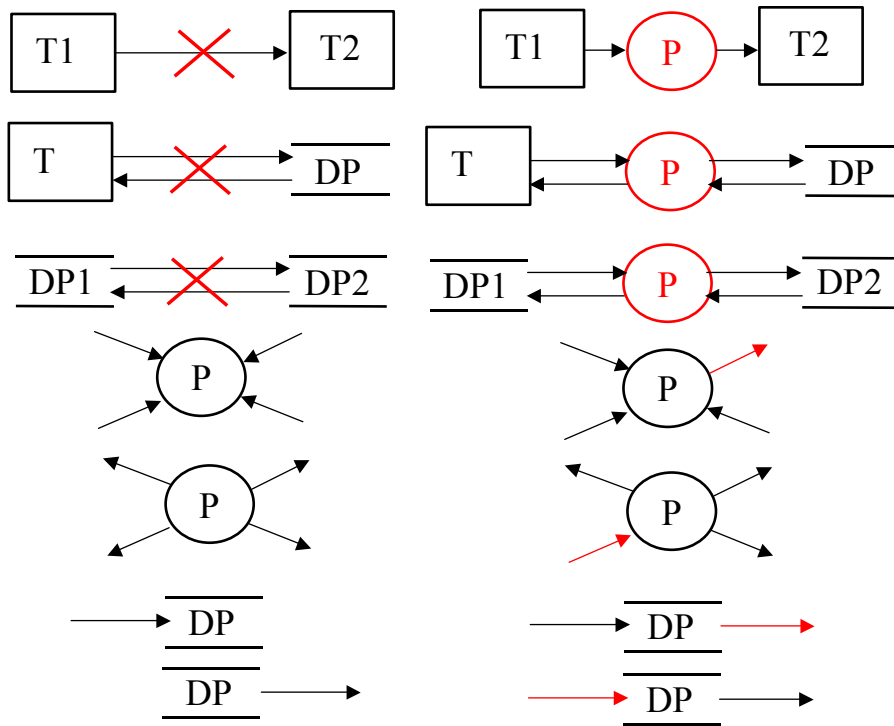
➤ **esenciální a implementační paměť**



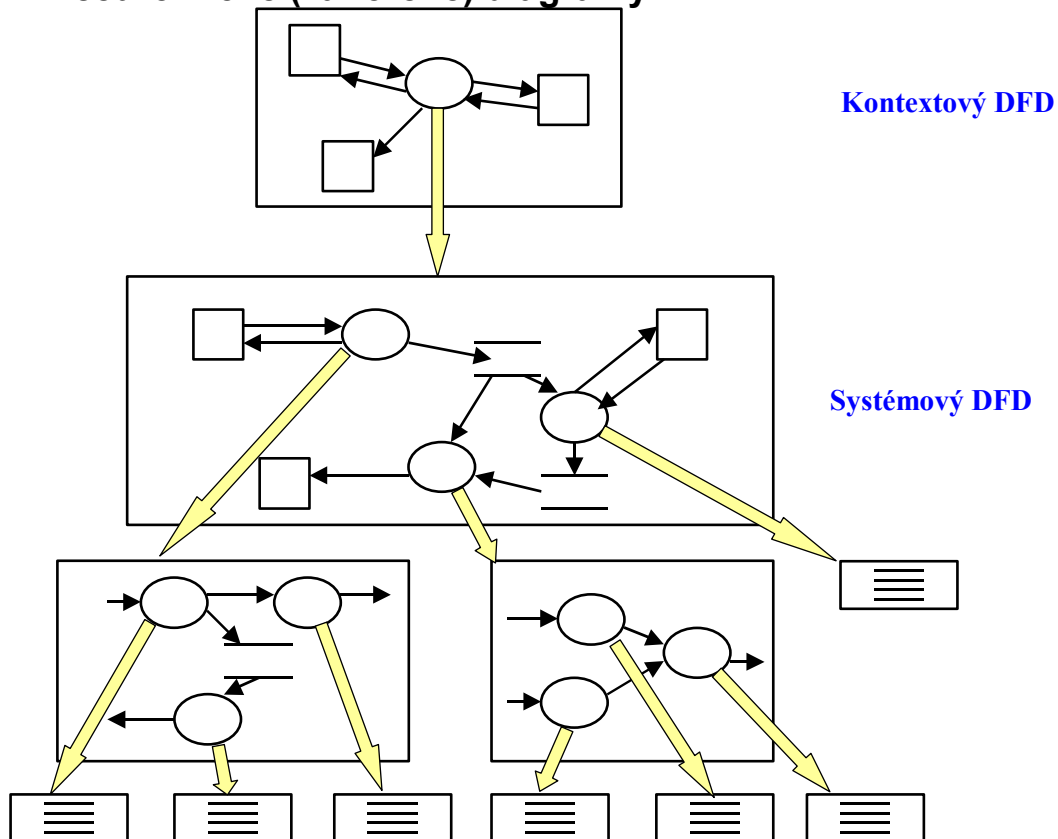
- **esenciální – požadavek uživatele**
- **implementační – spolehlivost, rozhraní podsystémů, ...**

• **Doporučení pro tvorbu DFD (Yourdon):**

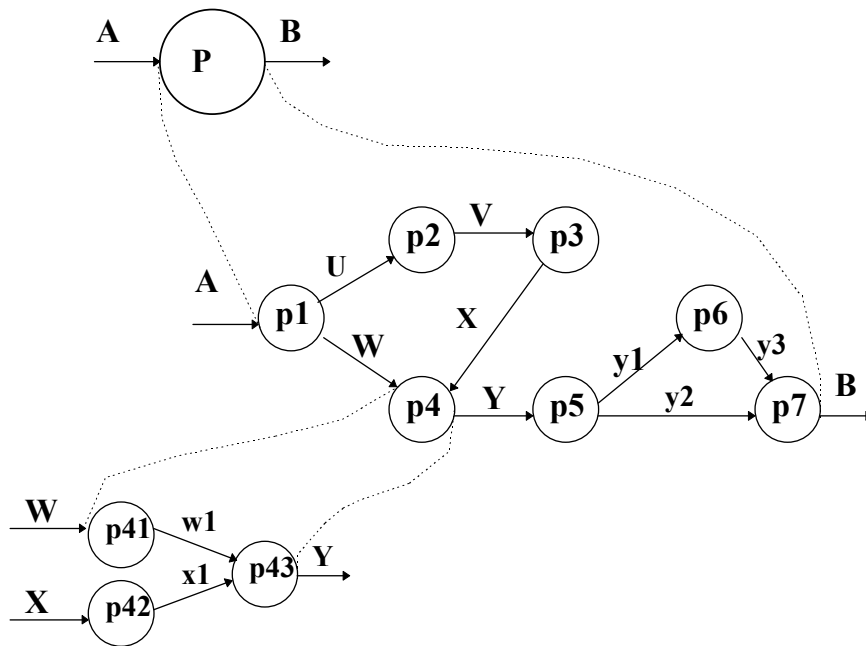
- **volit výstižná jména**
- **systematicky číslovat procesy**
- **tvořit estetické diagramy**
- **vyhnout se příliš jednoduchým i složitým diagramům**
- **ověřovat konzistenci**



• víceúrovňové (zanořené) diagramy



- **Pravidla tvorby víceúrovňových DFD**
 - **zachování kontinuity datových toků**



- **číslování procesů**
1, 1.1, 1.1.1, ..., jméno procesu je jménem DFD na nižší úrovni

- **umístění datových pamětí**
 - poprvé na nejvyšší úrovni, kde tvoří rozhraní procesů. Na nižší úrovni zpravidla opakujeme na diagramech procesů, které s ní spolupracují.
- **Další otázky při tvorbě a použití víceúrovňových DFD (Yourdon)**
 - **Jak určit počet úrovní DFD?**
 - ne více než 6 procesů na diagramu (formát A4), pro listové procesy jsme schopni napsat specifikaci procesu na jednu stránku.
 - **Existuje nějaký typický počet úrovní?**
 - pro jednoduché systémy 2 až 3 úrovně, pro středně velké 3 až 6 a pro rozsáhlé 5 až 8
 - **Musí být všechny části systému členěny na stejnou úroveň podrobností?**
 - **Jak je vhodné předvést DFD zákazníkovi?**

9.3 Datový slovník

- původně notace pro popis informačního obsahu prvků DFD a ERD, u CASE systémů obsahuje popis prvků modelů

Symbol	Význam	Příklad
=	skládá se z	I = RČ
+	a	J = KJ +P
()	volitelnost	J = KJ + P +(RJ)
[[]]	výběr	P = [muž žena]
{ } ⁿ	opakování	PSCČ = {číslo} ⁵
@	klíčová položka	Z = @OČ + J +
**	komentář	* toto je komentář*

```
jméno = (tituly) + @<2> křestní_jméno +
        (@<3>prostřední jméno + @<1>příjmení +
        (vědecká_hodnost)
tituly = {titul}
titul = [Prof. |Doc. |Bc. |Mgr. |Dr. Ing. | Ing. |dr. ]
vědecká_hodnost = [CSc. |DrSc. |PhD] ...
```

9.4 Specifikace procesů (minispecifikace)

- popis procesu, který není popsán jiným DFD. Použitelné rovněž pro popis algoritmů, operací objektů atd.
- popis procedurální nebo neprocedurální
- možnosti specifikace:
 - ◆ slovní popis v přirozeném jazyce,
 - ◆ strukturovaný přirozený jazyk,
 - ◆ strukturovaný jazyk (strukturovaná angličtina (čeština)),
 - kombinace jednoduchých vět, výrazů a řídicích struktur
 - slovník: imperativní slovesa, pojmy definované v datovém slovníku, klíčová slova řídicích struktur:
 - větvení (IF – THEN – ELSE, DO CASE – CASE – OTHERWISE),
 - opakování (REPEAT – UNTIL, DO WHILE, FOR EVERY – DO)

```
Zobraz fakturované objednávky
FOR EVERY objednávka v Objednávky
s datum-fakturace = dnešní datum DO
    ZOBRAZ číslo-faktury, jméno-zákazníka, částka
    denní-částka = denní-částka + částka
ENDFOREVERY
ZOBRAZ denní-částka
```

◆ pseudokód, jazyky pro popis programu (PDL),

```
Specifikace řízení pro ATM
procedure ATM is
  pin: Číslo_PIN;
  č_úctu: Číslo_úctu;
  stav: Množství;
  služba: Dostupné_služby;
  platná_karta, platný_PIN: Boolean;
begin
  loop
    Načti_kartu(č_úctu, pin, platná_karta)
    If platná_karta then
      Ověř_PIN(PIN, platný_PIN) ...
  end
```

◆ grafické notace (vývojový, Nassi-Schneiderman, Jacksonův diagram),

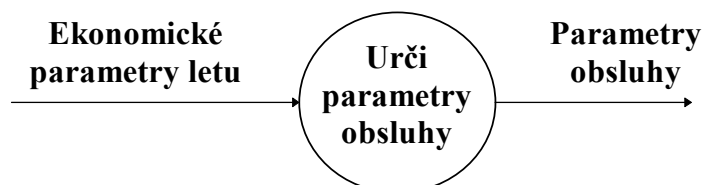
◆ pre-, post-podmínky,

```
procedure Search (Key: ELEM; T: ELEM_ARRAY; Found:
out BOOLEAN; I: out ELEM_INDEX);
Pre-condition
  TFIRST <= TLAST -- pole má alespoň jeden prvek
Post-condition
  -- prvek je nalezen a určen indexem I
  Found and T(I) = Key)
or
  -- prvek v poli není
  (not Found and not (exists i, TFIRST>= i <= TLAST,
  T(i) = Key))
```

◆ matematický popis, rovnice,

◆ rozhodovací tabulky, stromy

Př)

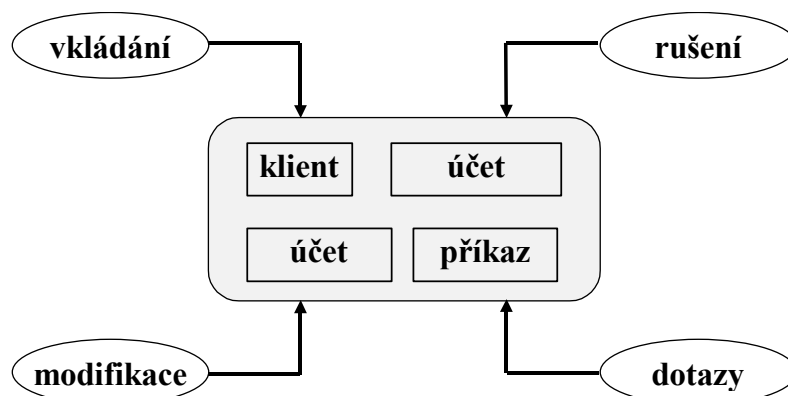


Proces Urči parametry obsluhy přijímá informace o ekonomických parametrech letu (charakter, obsazenost, průměrná cena letenky) a na jejich základě určuje parametry obsluhy podle těchto pravidel: Je-li let obsazen z více jak poloviny a průměrná cena letenky přesahuje 350\$, podávají se koktejly zdarma, nejde-li o vnitrostátní let. U vnitrostátního letu se podávají všechny koktejly, pokud je let obsazen více jak z poloviny. V takovém případě se všechny koktejly účtují.

Podmínky								
Vnitrostátní let	A	N	A	N	A	N	A	N
Obsazen nad 50%	A	A	N	N	A	A	N	N
Cena > 350\$	A	A	A	A	N	N	N	N
Akce								
Servírovat koktejly	A	A	N	?	A	?	N	?
Bezplatně	N	A			N			

9.5 Datové modelování – ER model

- slouží k modelování dat aplikační domény a jejich vztahů „v klidu“ – *konceptuální modelování*
- podstatné v datově intenzivních aplikacích



- ER modely (Entity Relationship), grafická forma - ER diagramy
- Která data potřebujeme v systému uchovávat? Jaké jsou mezi nimi vztahy?

- **Základní pojmy**

- **Entita** – „věc“ reálného světa (objekt) rozlišitelný od jiných objektů.

Př) Klient banky s identifikačním číslem 999, účet s č. účtu 100.

- **Entitní množina** - množina entit téhož typu, které sdílí tytéž vlastnosti (atributy).

Př) Klient, Účet

- **Atribut** - vlastnost entity, která nás v kontextu daného problému zajímá.

Př) Klient: čísloKlienta, jméno, příjmení, adresa, ...

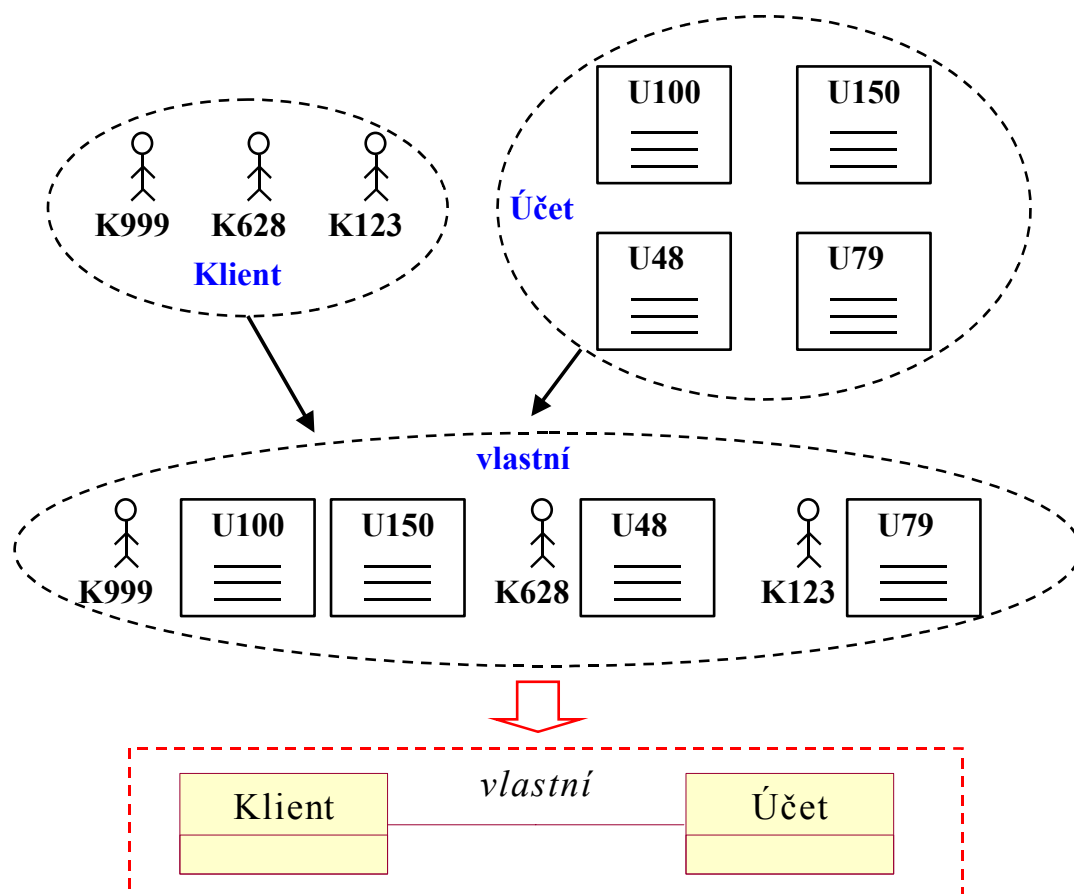
- **Vztah** – asociace mezi několika entitami.

Př) Klient s číslem klienta K999 vlastní účet s číslem účtu U100.

- **Vztahová množina** - množina vztahů téhož typu, které sdílí tytéž vlastnosti.

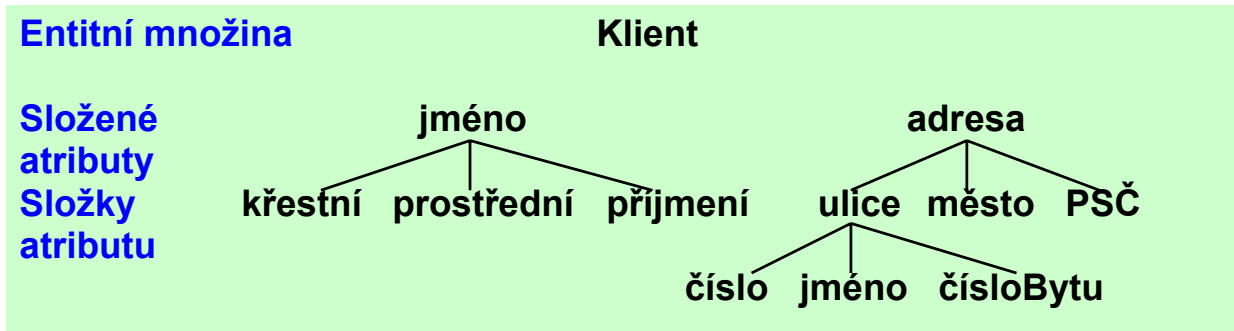
Př) Klient vlastní Účet – pro vztah mezi entitami typu Klient a Účet

Pozn.: Někdy také entita, resp. instance entity ve významu entitní množiny, resp. entity. Analogicky pro vztahové množiny a vztahy.



- Typy atributů

- Jednoduché (simple) a složené (composite) atributy



- Jednohodnotové (single-valued) a vícehodnotové (multiple-valued)

Př) telefon – může-li být několik čísel

- lze omezit minimální a maximální počet hodnot

- Prázdné (null) atributy

- mohou nabývat speciální „hodnoty“ NULL

- různý význam:

- ◆ chybějící - existuje, ale neznáme

- ◆ neznámá - nevíme, zda existuje

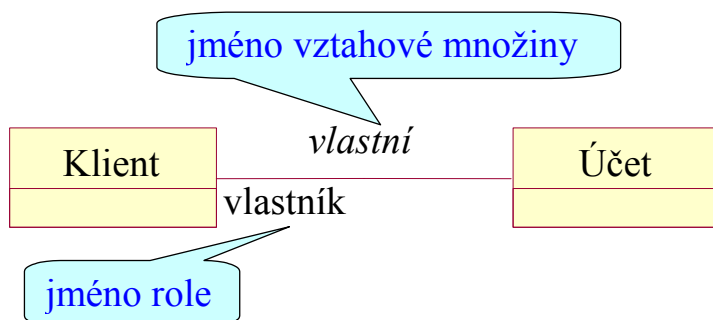
- Odvozené atributy

- hodnotu lze odvodit od jiných atributů nebo entit

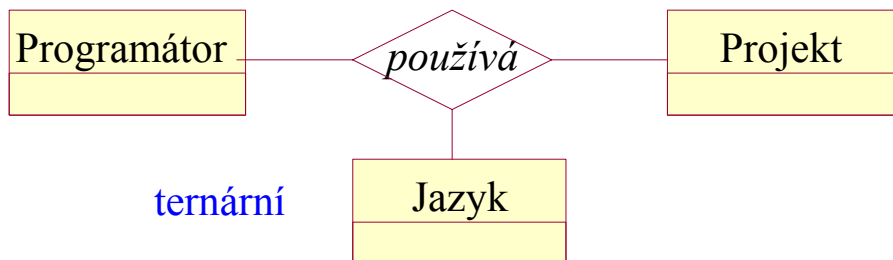
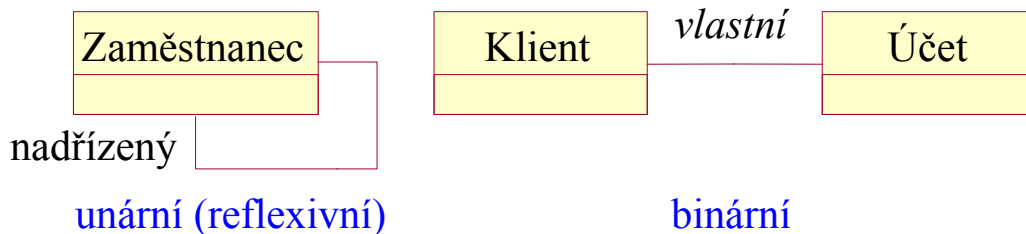
Př) věk, datNarození; početDispOsob

- Parametry vztahů

- Jméno vztahové množiny, jméno role – vyjadřuje význam vztahu

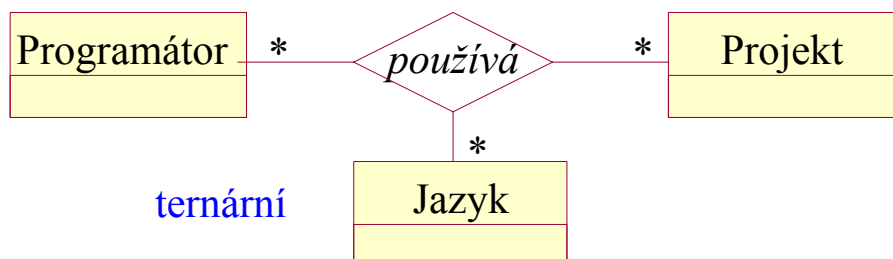
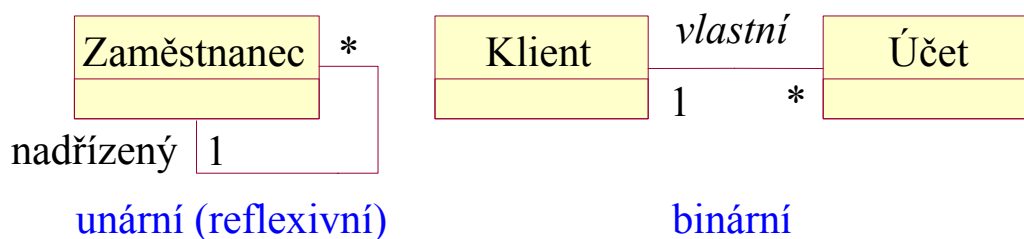


➤ **Stupeň**



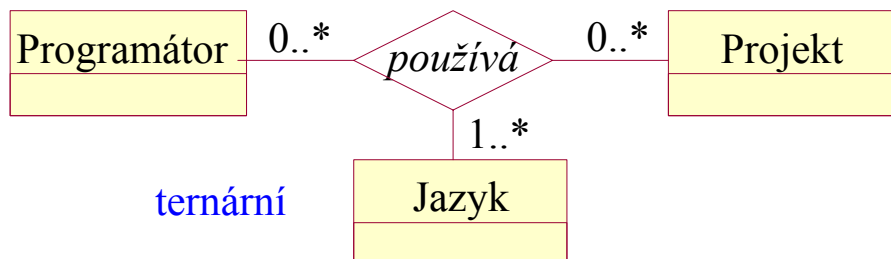
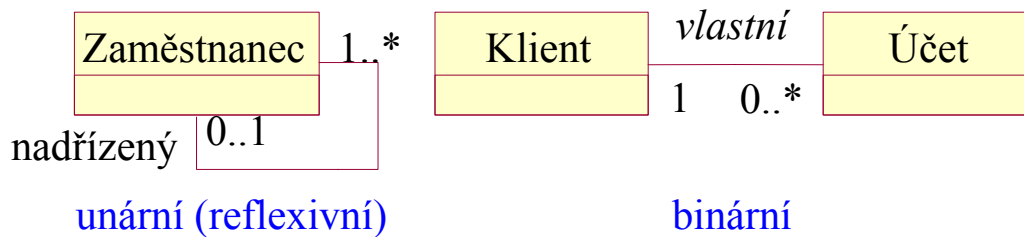
➤ **Kardinalita (cardinality),**

- maximální počet vztahů daného typu (vztahové množiny), ve kterých může participovat jedna entita (1,M, případně přesněji).



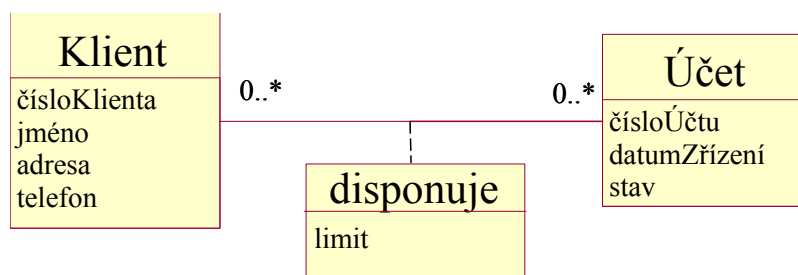
➤ **Členství (membership)/účast (participation)**

- minimální počet vztahů daného typu (vztahové množiny), ve kterých musí participovat jedna entita (0 – volitelné, 1 – povinné).
- také účast entitní množiny ve vztahové částečná (partial)/úplná (total).

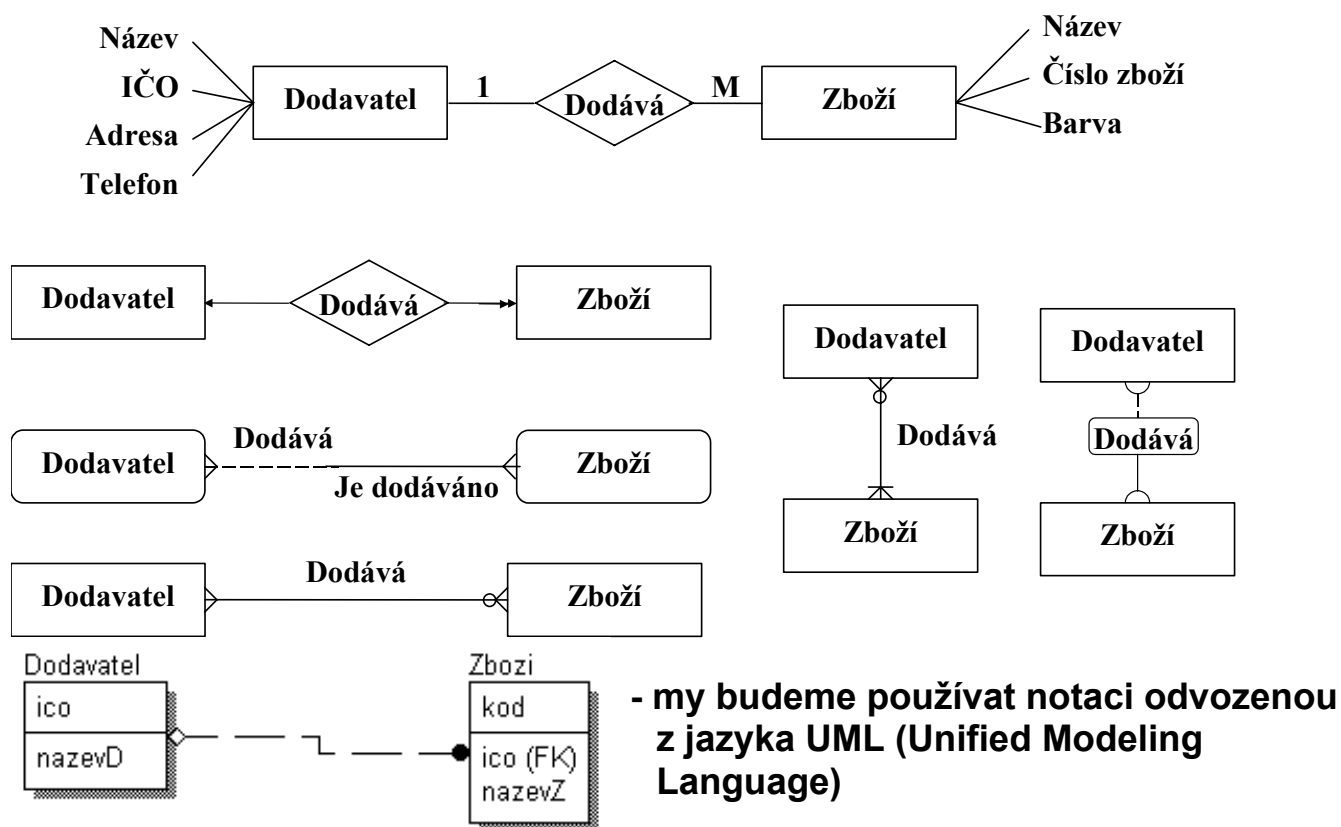


- kardinalita i členství představují omezení (constraint)

➤ **Atributy vztahu**



• Používané notace

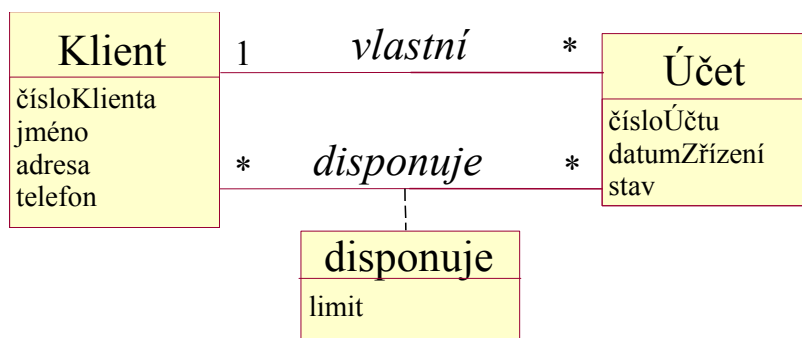


• Doporučení pro kreslení ERD

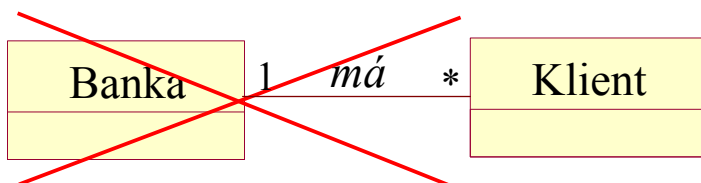
➤ Jména

- ◆ srozumitelná, musí vyjadřovat význam entitních a vztahových množin
- ◆ entitní množiny: podstatná jména
- ◆ vztahové množiny: slovesa, předložky
- ◆ je-li jméno vztahové množiny jasné ze jmen entitních množin, není nutné uvádět

➤ Několik různých vztahových množin mezi stejnými entitními



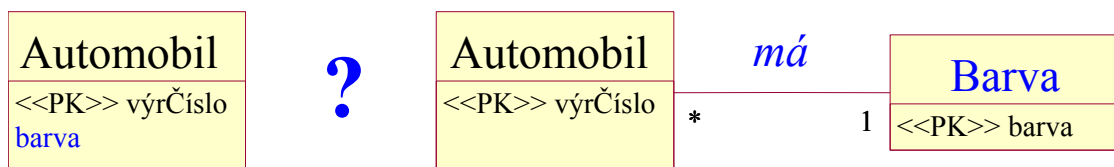
➤ Celkový systém by neměl být zahrnut do ERD



➤ Identifikátor (klíč, primární klíč)

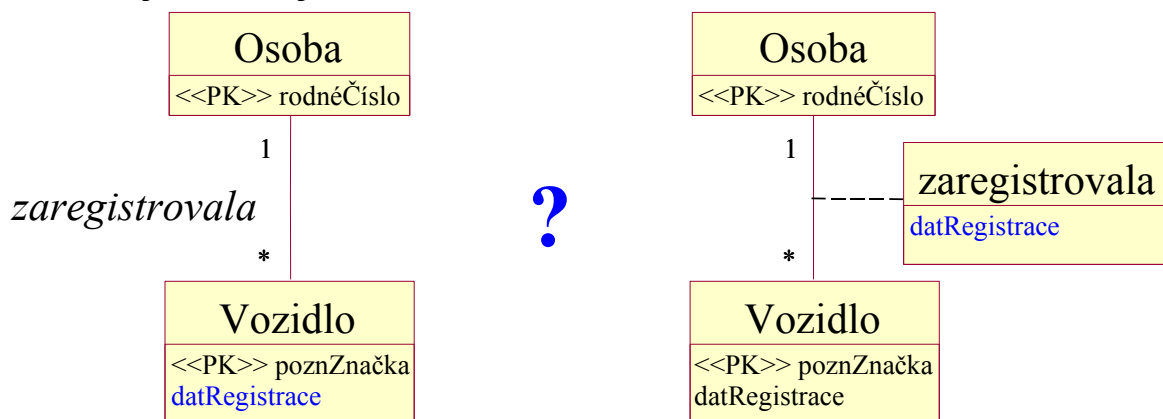
- ◆ entity a vztahy musí být identifikovatelné
- ◆ hodnota identifikátoru musí být unikátní (a minimální)
- ◆ identifikátorem je jednoduchý nebo složený atribut
- ◆ situace, kdy používáme složené identifikátory:
- ◆ unikátnost hodnoty jen v rámci vyvíjeného systému (ne celého vesmíru)

➤ Entitní množina nebo atribut?

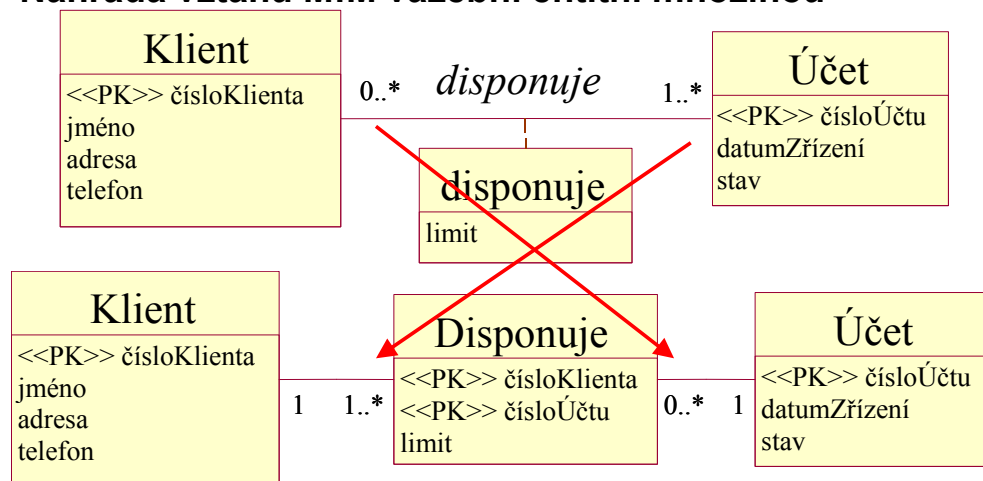


Pravidlo: Je-li hodnota atributu důležitá, i když neexistuje žádná entita s touto hodnotou jako vlastností, pak bychom ji měli modelovat jako entitu.

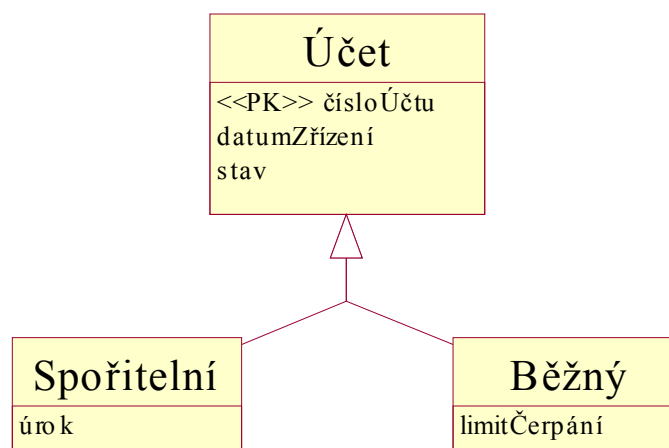
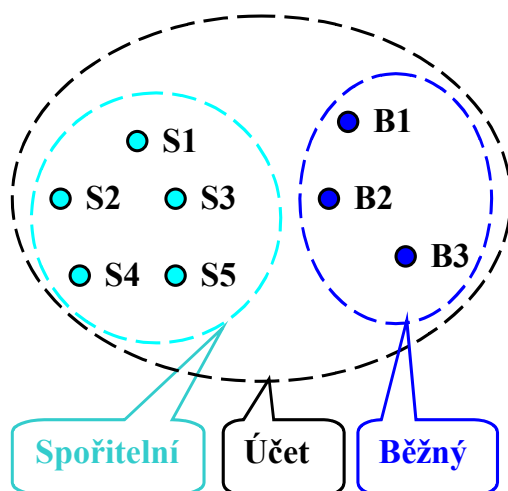
➤ Atributy a vztahy 1:M



➤ Náhrada vztahů M:M vazební entitní množinou

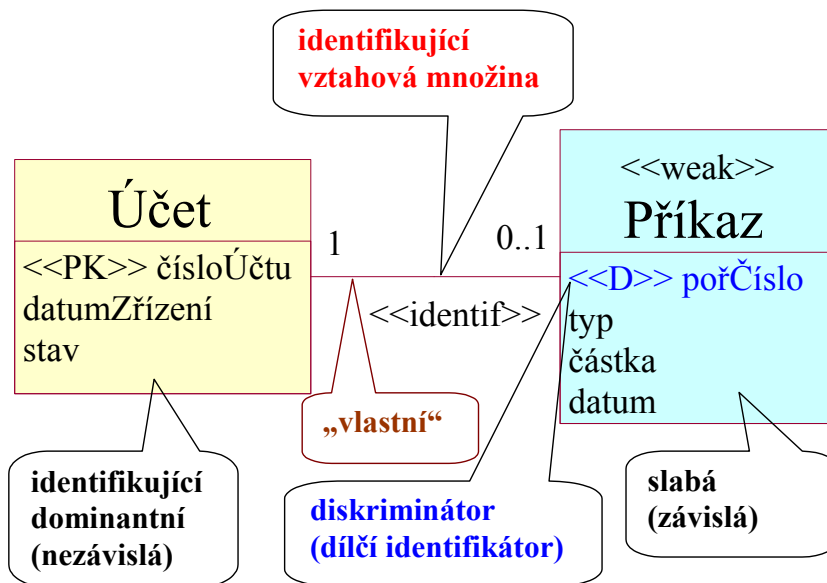


• Generalizace/specializace



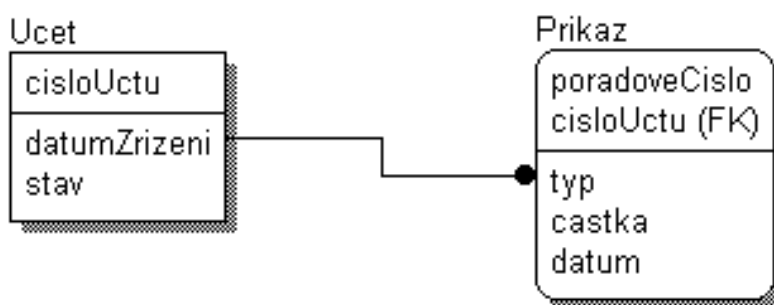
- také ISA vztah
- pojmy entitní množina vyšší/nížší úrovně (také nadtřída/podtřída)
- dědičnost atributů a účasti ve vztahových množinách
- hierarchie/svazy (lattice) generalizace
- identifikátor entitních množin nižší úrovně

- Slabé (weak) entitní množiny
 - *silná (strong) entitní množina* – má identifikátor tvořený vlastními atributy
 - *slabá entitní množina* – nemá identifikátor tvořený vlastními atributy



◆ Rysy slabé entitní množiny:

- o *identifikátor = identifikátor_dominantní + diskriminátor*
- o *existenční závislost slabé na identifikující*



◆ Slabá nebo silná entitní množina?

Pravidlo1: *Jako slabou modelovat tehdy, kdy entita kompletně zmizí při odstranění odpovídající identifikující entity.*

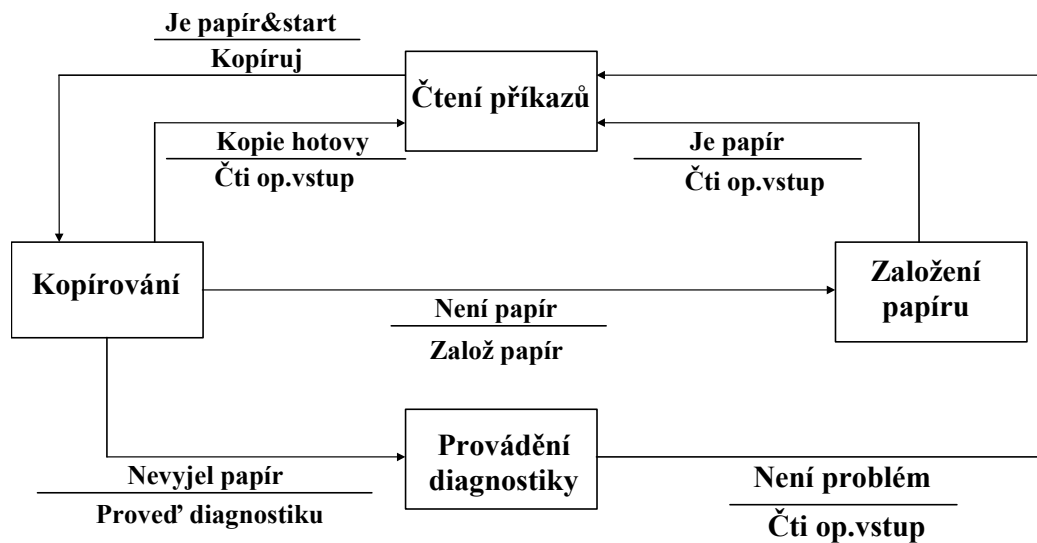
Př) Objednávka – PoložkaObjednávky

Pravidlo2: *Cokoliv s atributem, který je jednoznačný, by nemělo být modelováno jako slabá entitní množina.*

Pravidlo3: *Jsme-li na pochybách, modelujeme jako silnou entitní množinu.*

9.6 Stavový diagram

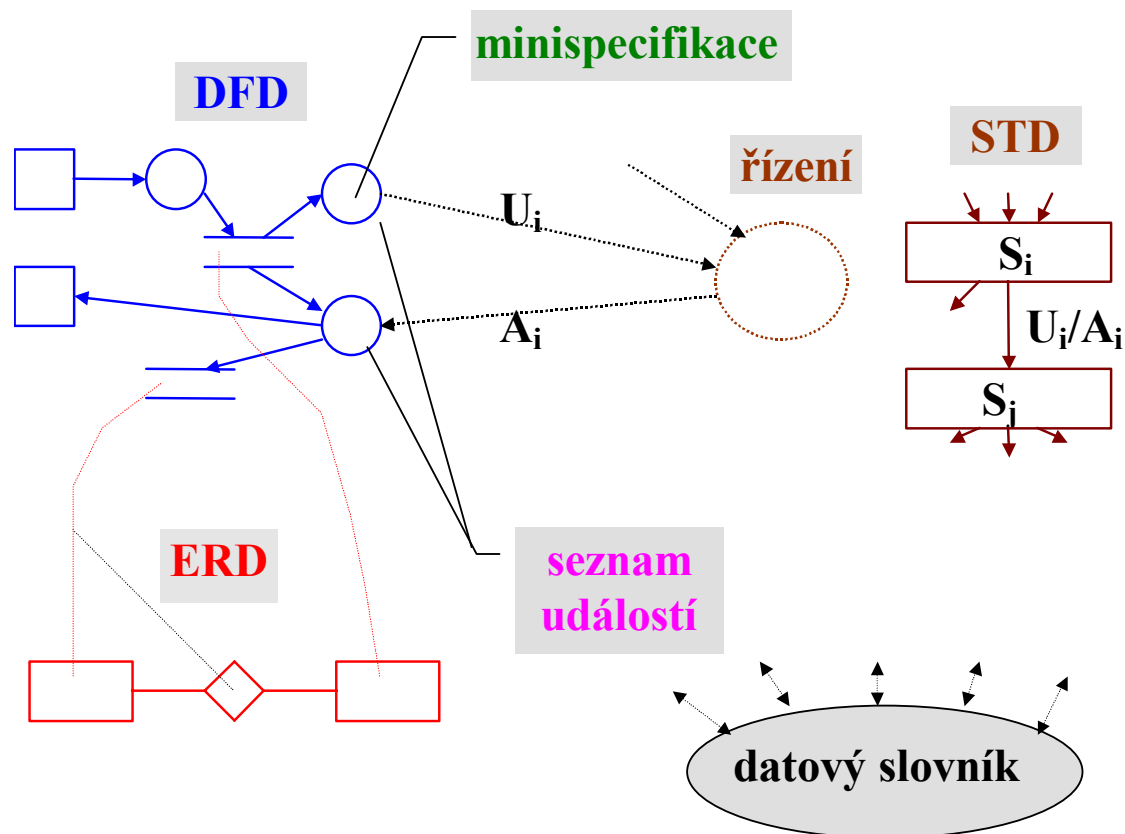
- pohled na systém nebo jeho část z hlediska stavů, přechodů, událostí, akcí. Často se používá k modelování řízení.



9.7 Další modelovací techniky

- Seznam vnějších událostí
 - **vnější události působící na systém, v systému musí existovat proces zodpovědný za zpracování události**
 - v Yourdonově Moderní strukturované analýze prostředek pro tvorbu prvotního DFD
- Matice CRUD, ...

9.8 Vztah modelů

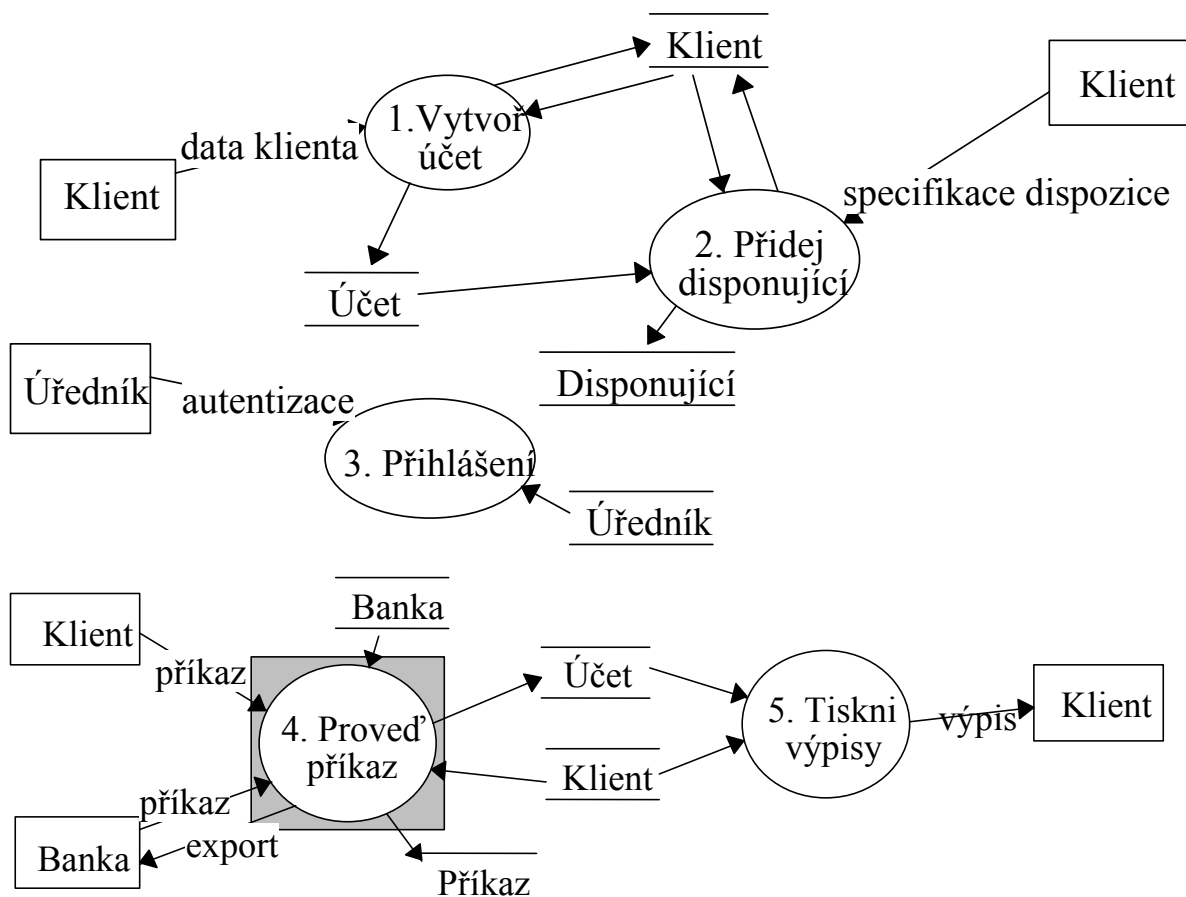
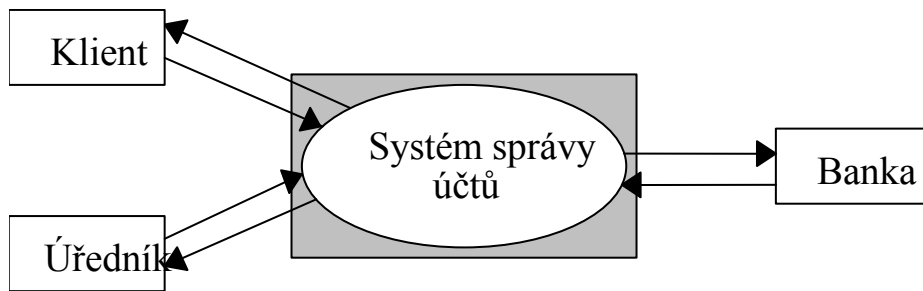


Př) Systém správy účtů

Provádíme analýzu systému správy účtů banky. Každý účet má jednoznačné číslo, dále je potřeba znát jméno a adresu majitele účtu. Kromě majitele mohou s účtem disponovat i další jím určené osoby. O těch je třeba znát stejné údaje jako o majiteli. Každá z disponujících osob může mít stanoven limit pro výběr z daného účtu. S účty manipuluje úředník banky na základě příkazu osoby oprávněné s účtem disponovat.

Na účet lze provádět vklad, z účtu lze provádět výběr a lze převádět částky na jiné účty v téže nebo jiné bance. Musí být k dispozici informace, kdo příkaz zadal a který úředník ho provedl. Systém musí poskytovat prostředky pro správu informací o klientech banky, musí umožňovat vytvářet a rušit účty, zadávat příkazy, importovat příkazy pro převody z jiných bank a naopak exportovat příkazy pro převody na účty v jiných bankách. Systém musí být schopen tisknout měsíční výpisy z účtů a řadu dalších tiskových sestav.

• DFD (ne všechny procesy jsou ukázány, dva dílčí systémové DFD)



• ERD

