

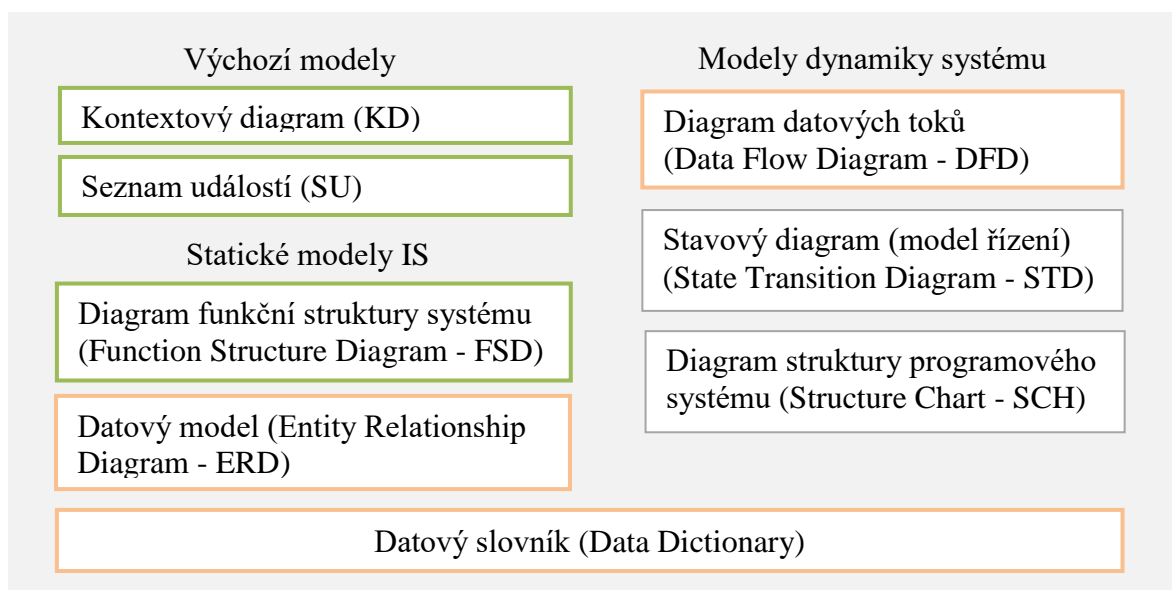
Předmět:	Projektování IS
Téma:	Strukturovaný přístup k vývoji IS (část B)

Vyučující:	dr. Dušan Kajzar	Školní rok:	2020/2021
------------	------------------	-------------	-----------

Obsah:

1. Diagram datových toků (DFD)	2
2. Datový model (ERD)	7
3. Datový slovník (Data Dictionary - DD)	14

Opakování - „Modely strukturovaného přístupu ...“



1. Diagram datových toků (DFD)

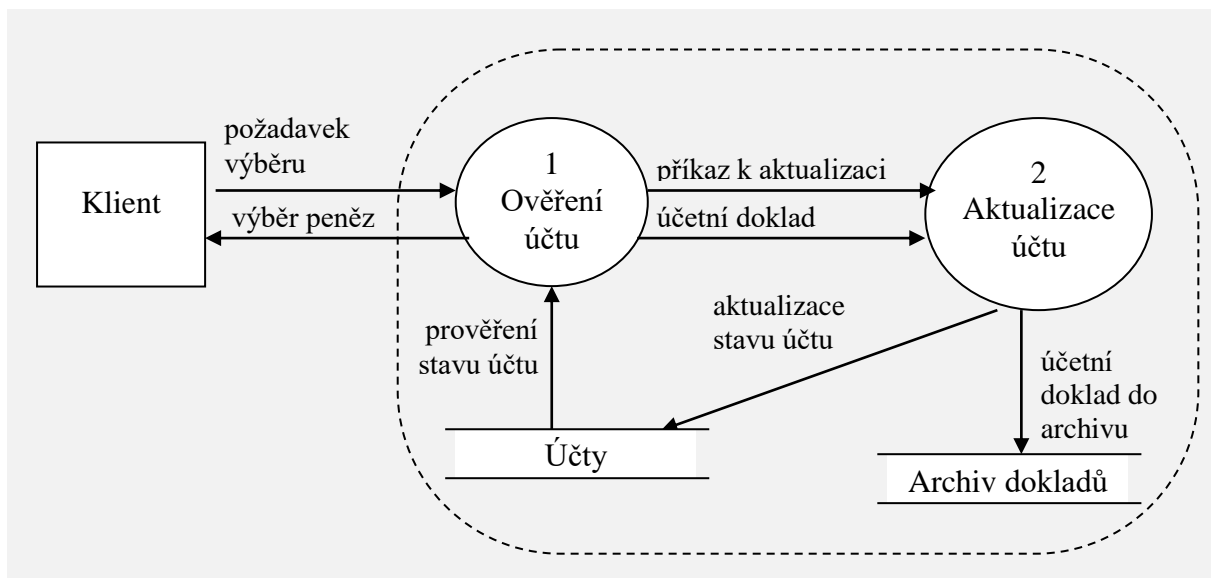
Účel modelu:

- zobrazit **dynamický pohled** na vyvíjený IS
 - procesy zpracovávající data,
 - datové toky mezi jednotlivými procesy,
- znázornění
 - interních **procesů** v systému,
 - **datových toků** mezi procesy,
 - **vstupních a výstupních** datových toků vzhledem k systému.

Grafické znaky - viz obrázek:

- procesy, datové toky (Data Flow), úložiště dat (Data Store),
- externí entity (terminátory).

Ukázka (fragment) diagramu datových toků - výběr z účtu:



Poznámka – co DFD nezobrazuje:

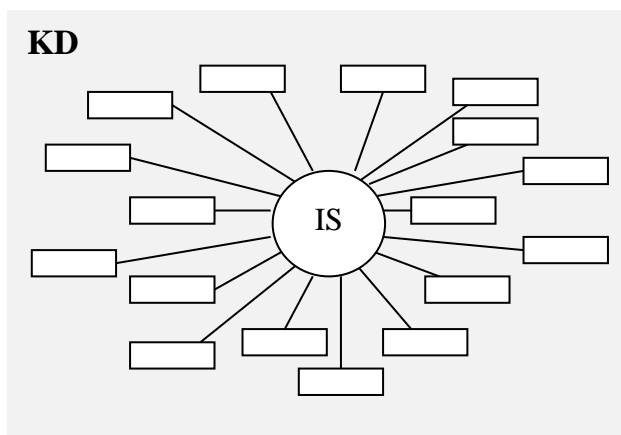
- časovou návaznost mezi procesy resp. datovými toky,
- tj. nevidíme zde pořadí datových toků a procesů,
- zobrazená vazba mezi procesy je čistě datová, nikoliv časová.

Bližší vysvětlení DFD:

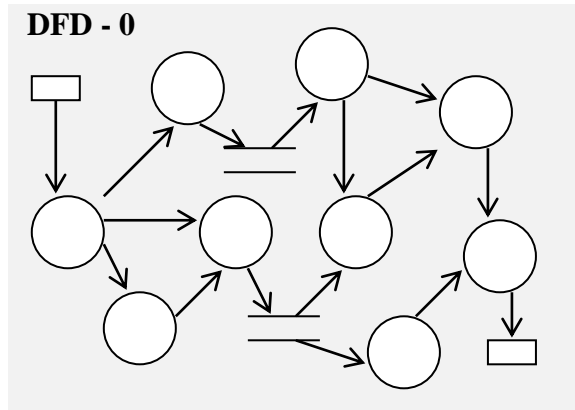
- vlastnosti procesů
 - hierarchické číslování,
 - vhodné pojmenování,
 - procesy datové a řídicí,
- vlastnosti datových toků,
 - data v pohybu (pohyb informací, materiálu, peněz, ...),
 - abstrakce od konkrétního způsobu přenosu dat,
 - datový tok do úložiště (insert, update, delete),
 - datový tok z úložiště (select),
- vlastnosti datových úložišť
 - množina dat „v klidu“,
 - abstrakce od konkrétního způsobu uložení dat,
 - data podniku + pracovní data (mezivýsledky),
 - pozor „pasivní prvek“ – práce s daty vždy vykonává proces,
- vlastnosti externích entit
 - prvky mimo náš vyvíjený systém,
 - důležité pro tvorbu rozhraní „systém – okolí“,
 - vazby mezi externími entitami – nás již nezajímají.

Hierarchie DFD:

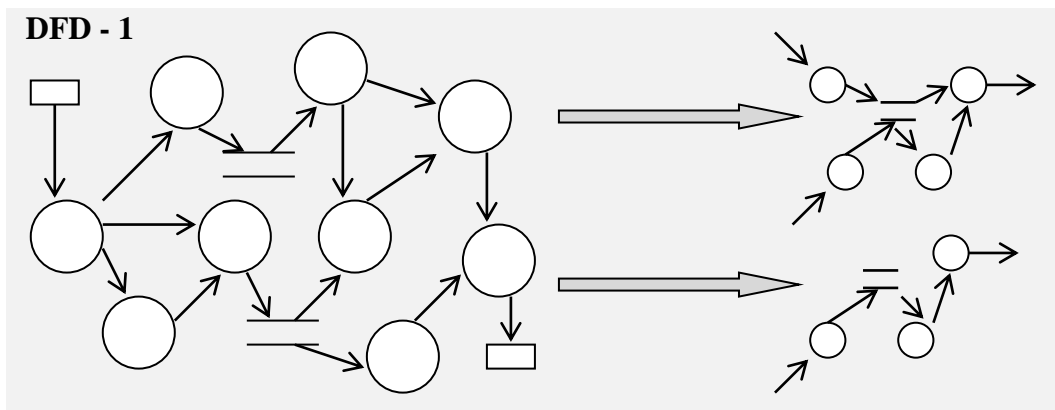
- zcela na vrcholu – Kontextový diagram,



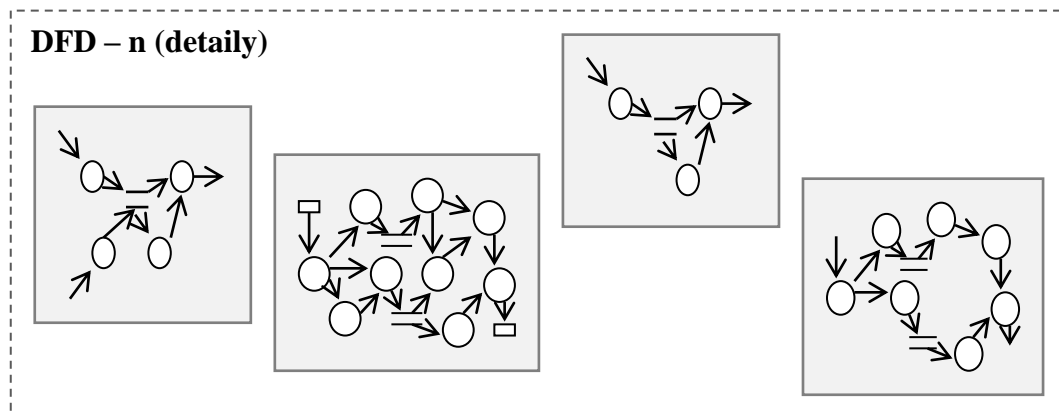
- nejhrubší úroveň zobrazení DFD – základní, nultá
 - vidíme hlavní subsystémy a vazby mezi nimi,
 - na dané úrovni - doporučeno zobrazit 3-9 subsystémů,
 - hierarchické číslování procesů,



- zobrazení na vyšších rozlišovacích úrovních
 - pohled na dekomponovanou část „zvětšovací sklem“,



- postupná dekompozice (rozklad)
 - procesů,
 - datových toků,
 - datových úložišť,
- zobrazení datových toků a úložišť
 - datové toky a úložiště jsou „ukryté“ v procesech,
 - viditelné až při rozpracování procesu na dané úrovni podrobnosti,
- dbát na konzistenci mezi úrovněmi zobrazení,
- rozklad až na tzv. „elementární procesy“ (viz dále)
 - dostatečná podrobnost pro implementaci.



Poznámky k zobrazení DFD:

- formální požadavky
 - např. „hlavička“ modelu (technického výkresu),
- zobrazení procesů
 - účelnost (užitečnost) zobrazovaných procesů,
 - estetický vzhled diagramů,
 - procesy vs. datové toky,
- zobrazení datových úložišť
 - datová úložiště vs. datové toky,
 - atributy datových toků a úložišť.

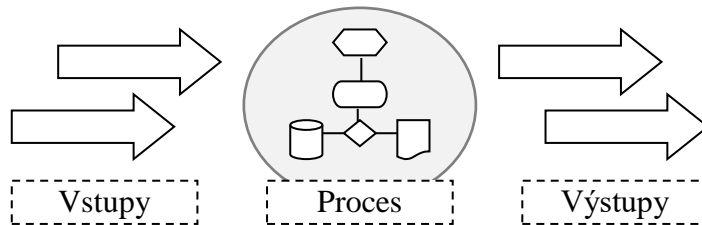
Řídicí proces:

- umožňuje vnést pohled návaznosti (posloupnosti) procesů a datových toků,
- **řídící pokyny**, impulsy, události
 - zahájení, ukončení datových procesů,
- doporučení - max. jeden řídicí proces v systému
 - je-li jich více – koordinovat jejich součinnost,
- úložiště událostí - **Event Store**,
- rozpracování algoritmu řídicího procesu v STD
 - v různých stavech systém generuje řídicí impulsy.



Elementární procesy:

- kdy proces považujeme za „elementární“?,
- popis elementárního procesu
 - tzv. minispecifikace (**Process specification**),
 - „hrubý“ algoritmus procesu, nejde o program (!),



- jazykové prostředky pro popis minispecifikací
 - vstupy, výstupy, sekvence, větvení, cyklus,
 - speciální strukturovaný jazyk,
 - vývojový diagram,
- srozumitelnost popisu
 - pro analytika, programátora, uživatele,
 - Proč pro uživatele?

Výsledek tvorby DFD:

- hierarchie dílčích modelů vyvíjeného IS.



Úzká souvislost DFD s jinými modely:

- souvislost s KD a SU,
- souvislost s ERD (viz dále),
- souvislost s STD (viz dále),
- souvislost s SCH (viz dále),
- souvislost s Data Dictionary (slovní popis prvků).

Rozdíl mezi DFD a FSD:

- FSD – statický pohled, hierarchická struktura subsystémů,
- DFD – dynamický pohled, procesy zpracování dat a datové toky mezi procesy.

Poznámka – souvislost s modely OOP:

- v OOP (UML 2.0) nemá přímo odpovídající model
 - do jisté míry lze nahradit Activity diagramem,
- v UML 2.5 byl doplněn Information Flow Diagram.

2. Datový model (ERD)

Účel modelu:

- zobrazit **strukturu dat** vyvíjeného IS,
- **entity**, **atributy** entit, **relace** mezi množinami entit.

Pojmy známé z předmětu „Databázové systémy“:

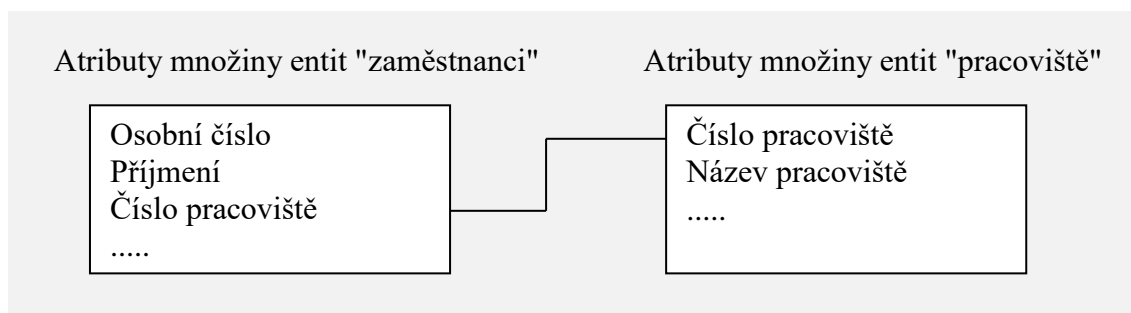
- entita, atributy entity, množina entit,
- primární klíč, cizí klíč, alternativní klíč,
- jednoduchý a složený klíč, nejednoznačný klíč.

Vazby mezi množinami entit:

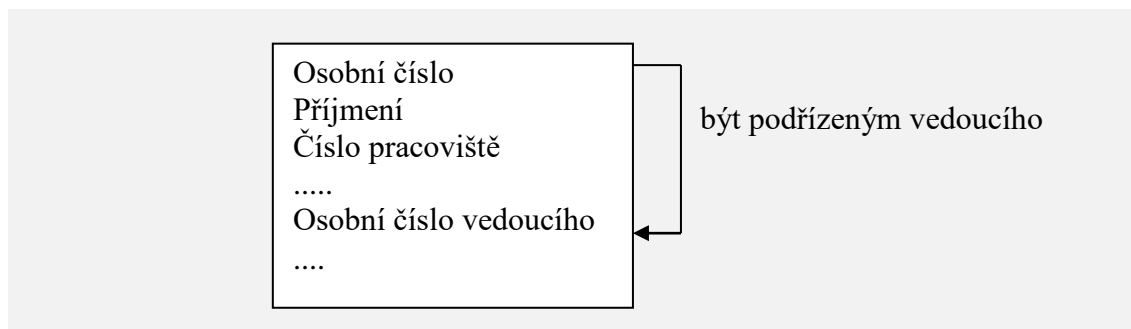
- stupeň vazby (unární, binární, ...),
- kardinalita (násobnost, 1:M),
- volitelnost (vazba povinná, volitelná).

Binární relace:

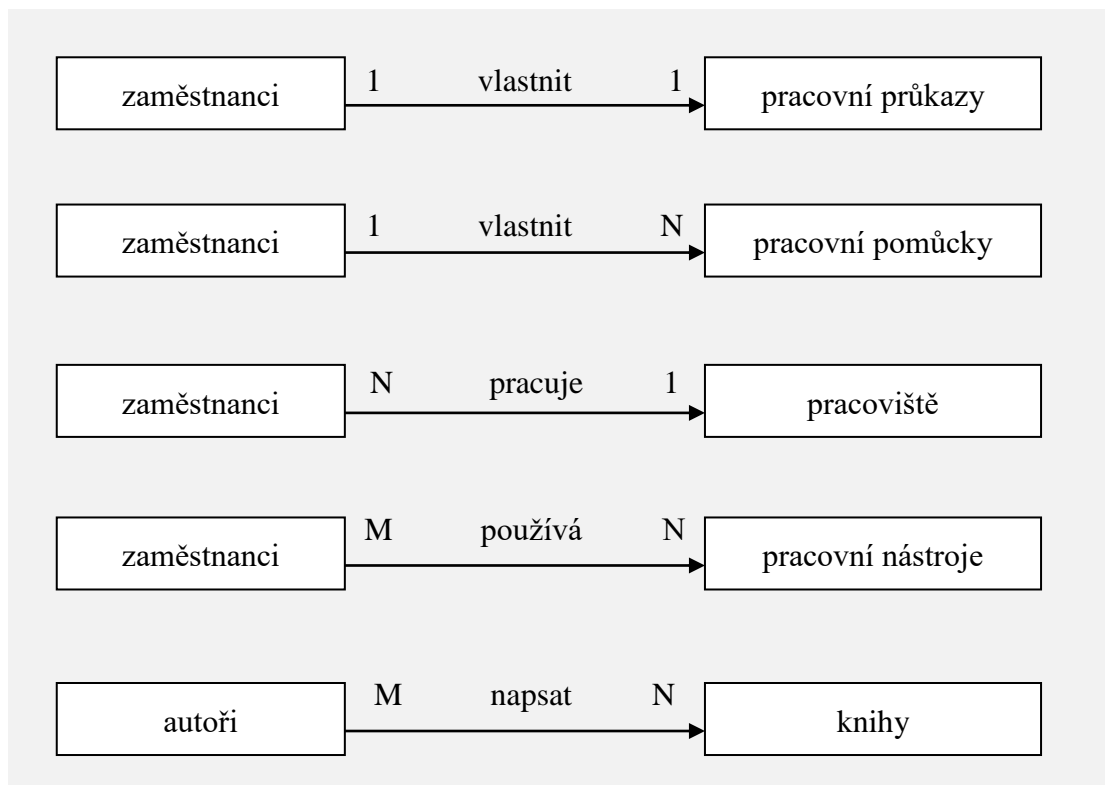
- obrázek - vazba množin entit přes cizí klíč "číslo pracoviště".



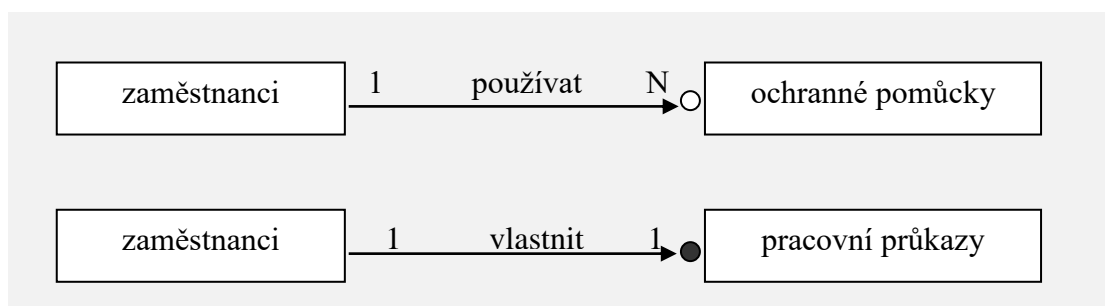
Unární relace:



Kardinalita (násobnost) relace:

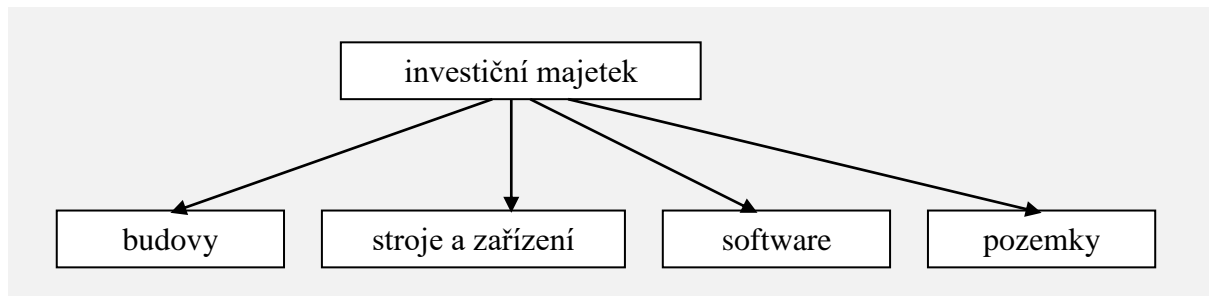


Povinnost a volitelnost relace:



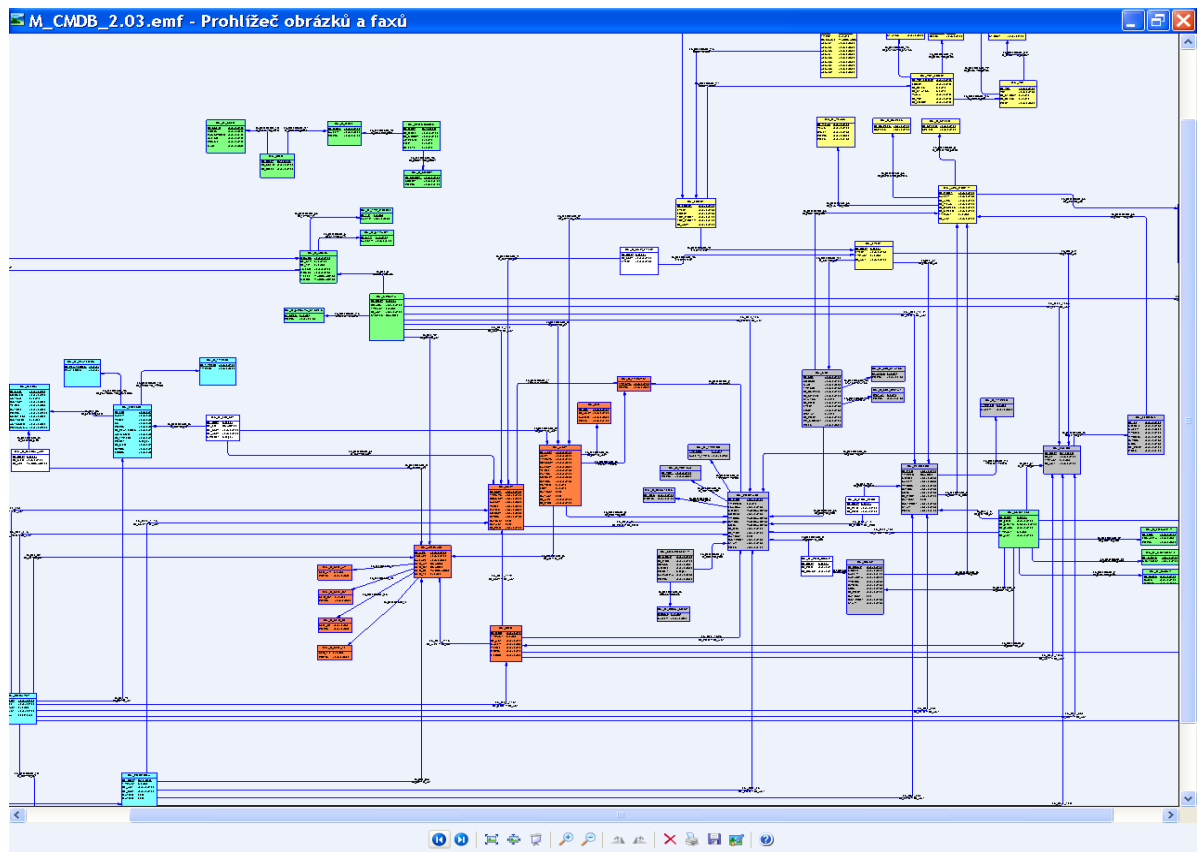
Generalizace a specializace množin entit:

- např. hlavní karta a podkarty investičního majetku.



Ukázka datového modelu:

- databáze CMDB (Configuration Management DB),
- poznámka - použití Sybase Power Designer Viewer.



Rámcový postup tvorby ERD:

- 1) výběr nejdůležitějších množin entit
 - např. zaměstnanec, pracoviště, útvary, povolání, mzdové složky, atd.,
- 2) určení základních atributů jednotlivých množin entit,
 - např. zaměstnanec – ID, jméno, příjmení, dat. narození, dat. nástupu, ...
- 3) určení primárního klíče jednotlivých množin entit,
- 4) průběžné zakreslování množin entit a vazeb do ERD,
- 5) vyhledávání vazeb (relací) mezi množinami entit,
- 6) doplňování dalších prvků do modelu
 - doplňování dalších entit,
 - doplňování dalších atributů,
- 7) určování kardinality a volitelnosti vazeb,
- 8) průběžná revize modelu, např.
 - odstranění nadbytečných množin entit a atributů,
 - odstranění zbytečných vztahů mezi entitami,
- 9) normalizace datového modelu,
- 10) stanovení omezujících podmínek pro hodnoty atributů
 - integritní omezení,
- 11) kompletace hierarchie množin entit
 - generalizace a specializace, nadřízené a podřízené entity,
- 12) důkladné ověření úplnosti datového modelu,
- 13) doplňování a úpravy modelu na základě ověřování úplnosti.

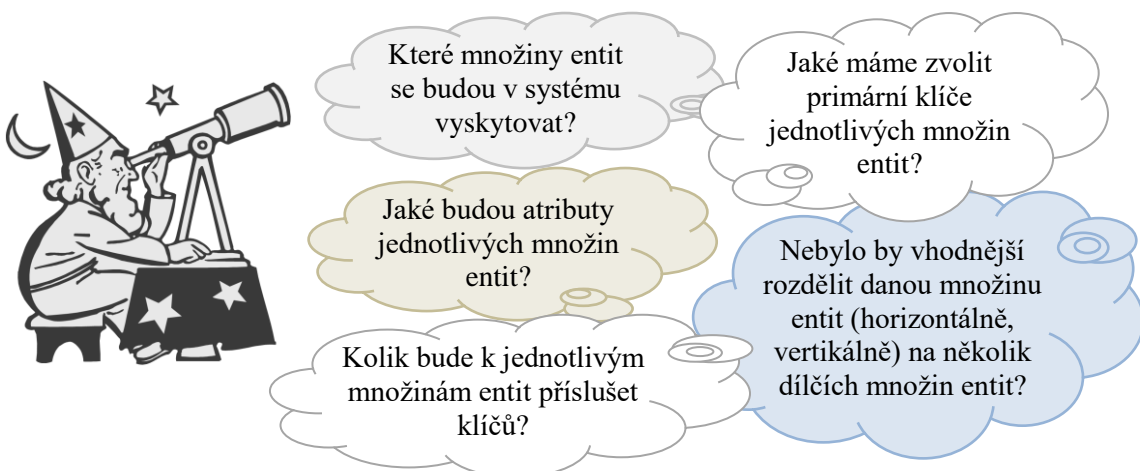
Opakování z předmětu „Databázové systémy“:

- normalizace modelu:
 - 1. NF - každý sloupec obsahuje atomické hodnoty,
 - 2. NF - 1. NF + každá neklíčová položka závisí na celém klíči,
 - 3. NF - 2. NF + neklíčová položka nezávisí na jiné neklíčové položce,
 - proces normalizace a denormalizace,
- integrita dat (celistvost, ucelenost, neporušenost):
 - integrita entity,
 - doménová integrita,
 - referenční integrita.

Ověření úplnosti datového modelu (!):

- Řeší model všechny potřebné úkony a SQL dotazy nad daty IS?
- např. Jak zaeviduji ... ?
 - že daná virtuální IP adresa balancuje nad několika logickými IP,
 - že logické IP adresy mohou migrovat mezi uzly v clusteru,
 - že logická IP adresa se váže k daným fyzickým IP adresám,
 - zachycení více síťových interfaces v serveru, IP multipathing, ...
 - apod.
- např. „Jak vypíši ... ?“
 - všechny servery s končící dobou supportu k 31.12. v lokalitě Vítkov,
 - všechny db systémy Oracle a jejich licenční pokrytí,
 - všechny servery Linux, jejich umístění v serverovně a racku, tříděné podle účelu serveru (databázový, aplikační, ...),
 - na kterých db serverech pracuje aplikace „platební styk“, včetně jejich umístění v lokalitě, serverovně, racku, odpovědný správce, ...
 - ve kterých verzích provozujeme db Sybase, kdo tyto systémy spravuje a kde jsou instalační média jednotlivých verzí,
 - které systémy budou nepřístupné po výpadku switchu AX254,
 - přehled virtuálních IP adres a systémů, ke kterým se vztahují,
 - atd.

Příklady otázek, které je nutno řešit během tvorby ERD:



Je DB model dostatečně normalizovaný (ve 3. NF)?

Bylo by vhodné databázový model účelově denormalizovat?

Které množiny entit budou obsahovat data charakteru číselníků?

Které množiny entit budou obsahovat relativně stálá data, které pak data rychle se mění?

Na implementační úrovni:

Jakým způsobem zajistíme aplikační logiku (uložené procedury, triggers)?

Jakým způsobem zajistíme integritu dat (použitím constraints, triggers)?

Jakým způsobem zajistíme požadovanou bezpečnost systému ?

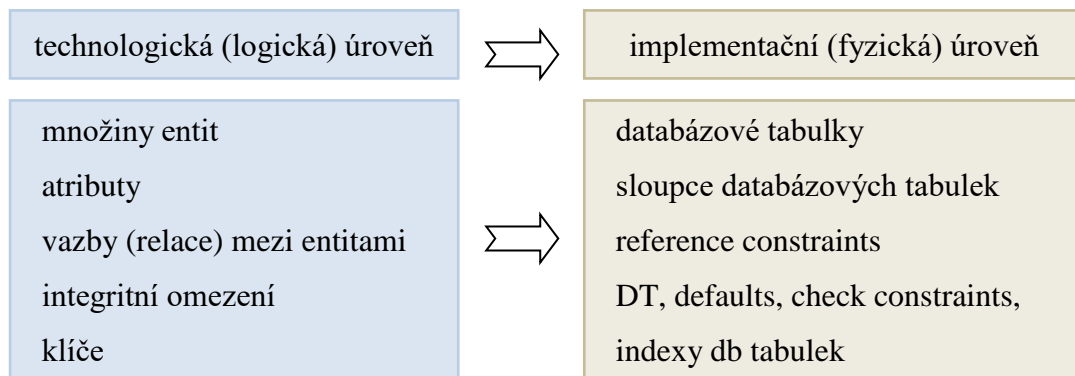
Poznámka ke kvalitě návrhu DB modelu:

- návrh datového modelu by měl garantovat „zkušený“ návrhář,
- nevhodný návrh DB
 - v praxi častou příčinou **výkonnostních problémů IS**,
- často se návrh ukáže nevhodný až **po jisté době** rutinního provozu
 - např. po zaplnění velkými objemy dat,
 - po doplnění další funkcionality systému (problém zamykání zdrojů),
 - tj. změní se provozní podmínky, pro které byl vytvořen (nemusí jít přímo o chybu návrháře),
- **ladění výkonnosti IS**
 - může vést k potřebě upravit datový model,
- **úpravy datového modelu** u IS v rutinním provozu
 - provozně náročné akce,
 - v databázi jsou již velké objemy dat (např. stovky GB, ...),
 - jak transformovat data do nové struktury?
 - kde aspoň dočasně alokovat diskový prostor (diskové pole je zaplněné)?
 - jak se vejít do provozně akceptovatelné doby odstávky systému?
 - apod.

Iterativní způsob tvorby modelu:

- datový model na konceptuální úrovni zobrazení,
- datový model na technologické (logické) úrovni zobrazení,
- datový model na implementační (fyzické) úrovni zobrazení.

Transformace zobrazení v ERD:



Souběžná tvorba ERD a DFD:

- paralelní vývoj a vzájemné doplňování modelů,
- iterativní postup.

Úzká souvislost ERD s jinými modely:

- souvislost s DFD,
- souvislost s SCH (viz dále),
- souvislost s Data Dictionary (slovní popis prvků).

Poznámka – souvislost s modely OOP:

- diagram tříd – Class diagram.
- ERD je nutný, pokud IS má mít relační databázi,
- + zpracování návrhu tzv. objektově relačního mapování.

3. Datový slovník (Data Dictionary - DD)



Účel modelu:

- popis prvků, které se v modelech IS vyskytují,
- významový slovník pojmů (prvků) v modelech IS,
- grafické modely IS jsou názorné, ale rovněž je potřebný popis.

Struktura popisu prvku:

- jednoznačný identifikátor prvku,
- význam prvku v rámci vyvíjeného IS,
- popis struktury prvku (složení z elementárních prvků),
- formát prvku a přípustné hodnoty.

Například:

- Objednávka vysvětlení významu,
- Objednávka popis struktury prvku,
- Objednávka = Hlavička objednávky + Tělo objednávky,
- Hlavička objednávky = Číslo obj. + ID zákazníka + Datum vystavení obj.,
- Číslo obj. = ABCXXXX, kde A=, B=, C=, XXXX=,
- atd.

Poznámky ke tvorbě DD:

- popisované prvky musí mít jednoznačné identifikátory,
- elementární prvky - charakterizuje typ dat a množina povolených hodnot,
- složené položky - popsat odkazem na elementární prvky,
- nutno dbát - na minimální redundanci informací v DD.

Použití CASE nástroje:

- ke tvorbě DD nás vede CASE nástroj,
- průběžně vede k zadávání údajů a jejich popisů,
- DD je pak součástí repository (tj. databáze) použitého CASE.