

# Projektování informačních systémů 5

---

**Příprava prováděcího projektu,  
strukturované a objektové  
metody analýzy a návrhu**

Doc. Mgr. Petr Suchánek, Ph.D.

Doc. RNDr. Ing. Roman Šperka, Ph.D.

Převzato od: Ing. Dominik Vymětal, DrSc.

# Příprava prováděcího projektu začíná detailní analýzou


---


- Analýza a návrh probíhá různými způsoby
- Vždy však zahrnuje analýzu požadavků uživatele
- Návrh probíhá s využitím různých přístupů
  - BPM + návrh
  - Známé metodologie
  - Agilní metodiky
  - Hodnotové přístupy k modelování
  - Metodiky SW firem

# Základní pojmy

---

- Metodika, metodologie
  - Doporučený souhrn přístupů, zásad, postupů, metod, technik atd. pro tvůrce IS
  - Kdo co a proč
- Metoda určuje co je třeba dělat v určité fázi
  - Přístupy jako funkční, datový, objektový
  - Řeší postup v určité fázi
- Technika – přesné postupy
- Nástroj – prostředek uskutečnění určité činnosti
  - Diagramy, atd.

Dimense IS/IT	METODA	TECHNIKA	NÁSTROJ 
DATA/ INFORMAC E	<ul style="list-style-type: none"> <li>-metoda datové analýzy</li> <li>-metoda funkční analýzy</li> <li>-objektově orientované</li> <li>-informační analýza (ISAC)</li> </ul>	<ul style="list-style-type: none"> <li>-technika datové analýzy (normalizace, kanonická procedura)</li> <li>-techniky návrhu logických datových struktur</li> </ul>	<ul style="list-style-type: none"> <li>-ER Diagram (konceptuální popis entit a jejich vztahů)</li> <li>-strukturní diagram</li> <li>-jazyk Data Dictionary</li> <li>-objektové modelovací nástroje (data objektů a jejich vztahů)</li> </ul>
FUNKCE/ PROCESY	<ul style="list-style-type: none"> <li>-metoda datové analýzy</li> <li>-metoda funkční analýzy</li> <li>-objektově orientované</li> <li>-informační analýza (ISAC)</li> </ul>	<ul style="list-style-type: none"> <li>-techniky funkční analýzy (inkrementální návrh)</li> <li>-techniky návrhu modulární struktury programového systému (transakční a transformační analýza)</li> </ul>	<ul style="list-style-type: none"> <li>-diagram datových toků</li> <li>-stavový diagram</li> <li>-nástroje popisu algoritmu (strukturní diagram)</li> <li>-objektové modelovací nástroje (dynamika vztahů)</li> </ul>
ORGANIZAČNÍ A LEGISLATIVNÍ ASPEKTY	<ul style="list-style-type: none"> <li>-Business System Planning</li> <li>-Informační analýza (ISAC)</li> </ul>	<ul style="list-style-type: none"> <li>-Maticové techniky BSP (matice odpovědností a organizačních míst, informační kříž)</li> </ul>	<ul style="list-style-type: none"> <li>- maticové techniky BSP – informační kříž</li> </ul>

Dimenze IS/IT	METODA	TECHNIKA	NÁSTROJ 
PRACOVNÍ, SOCIÁLNÍ A ETICKÉ ASPEKTY	-Business System Planning -Informační analýza (ISAC)	-Maticové techniky BSP (matice odpovědností a organizačních míst, informační kříž	maticové techniky BSP – informační kříž
SOFTWARE	-Business System Planning -Informační analýza (ISAC)	-Maticové techniky BSP (matice odpovědností a organizačních míst, informační kříž	maticové techniky BSP – informační kříž
HARDWARE	-Business System Planning -Informační analýza (ISAC)	-Maticové techniky BSP (matice odpovědností a organizačních míst, informační kříž	- maticové techniky BSP – informační kříž
EKONOMICKÉ A FINANČNÍ ASPEKTY	-Metody plánování a řízení projektu	Techniky plánování a řízení projektu	- Nástroje plánování a řízení projektu

# Detailní analýza požadavků na nový systém

---

- Cíl:
    - Na základě firemní strategie a cílů projektu definovat detailní požadované funkce
  - Forma:
    - Workshopy s řízenou moderací a dokumentací
  - Účastníci:
    - Dle témat jednotlivé specializované dílčí týmy
    - K integraci je nutné dílčí týmy koordinovat (úloha projektového manažéra)
  - Výstup:
    - Podklady pro návrh databáze
    - Podrobná specifikace funkcí a prací
    - Podklady pro cílový koncept řešení
-

# Principy analýzy a návrhu

---

Analýza a návrh systému se řídí několika principy:

- Princip abstrakce – vyvoláno nutností pracovat se zvládnutelnými úseky (částmi)
- Top down hierarchie
- Princip tří architektur
- Agregace – specializace – generalizace
- Princip modelování (formální zjednodušené zobrazení systému/reality)

# Popis očekávané reality

---

- Je základem pro tvorbu nového systému
  - Konkrétní forma – model
    - Znázornění stávajícího nebo vytvářeného systému
    - Specifikace struktury a chování systému
    - Specifikace omezení a vazeb na okolí
  - Tři nutné předpoklady
    - Notace – způsob zápisu modelu systému
    - Proces – způsob použití notace
    - Nástroj – tool k dokumentování našeho modelu
  - Forma
    - Top down kaskáda – viz princip top-down hierarchie ne vždy možná
    - Iterační a přírůstkový cyklus
-



# Základní přístupy k zobrazení očekávané reality

---

## □ **Strukturované**

- Člení projekt na definované substruktury
- Probíhají více či méně ex post balancováním funkčního a datového modelu
- Data Flow Diagram (DFD), Entity Relationship Diagram (ERD), State Transition Diagram (STD), Data Dictionary a Process Specification

## □ **Objektové**

- Princip je ve spojení dat a služeb – objekty volají metody jiných objektů
  - Zapouzdřenost chování a dědičnost
-

# Strukturované přístupy k zobrazení reality – základní charakteristika

---

- Člení projekt na malé dobře definované aktivity, určují jejich posloupnost a vazby (dekompozice)
- Používají techniky modelů a diagramů
- Používají balancování funkčního a datového modelu
- Svou názorností umožňují zapojení i méně zkušených pracovníků
- **Metodologie a metody**
  - Yourdon Structured Analysis (YSA) – klasika a základ pro další, Jacksonova strukturní metoda, SSADM
- **Technika**
  - Chenův model ERA pro modelování dat jako základ strukturovaných CASE TOOLS, Jacksonovy strukturní diagramy
- **Nástroj**
  - DFD a další diagramy, možnost nástrojů CASE pro strukturované metody

# Yourdon Structured Method YSM

---

- ❑ V 80. letech Yourdon vyvinul metodu Yourdon Structured Method (YSM), která je založená na funkčních strukturách. Metoda podporuje 2 fáze ve vývoji SW: **analýza a návrh**. YSM zahrnuje 3 diskrétní kroky: studie proveditelnosti, základní modelování a implementační modelování. Dále nabízí další modely:
  - ❑ **Model chování**: chování systému může být popsáno 3 způsoby: funkčně, dynamicky a vztahově.
  - ❑ Processor environment model (**PEM**): popisuje alokaci výpočtových funkcí v hardwaru procesoru.
  - ❑ Software environment model (**SEM**): definuje softwarovou architekturu a její dopady na každý procesor.
  - ❑ Code organizational model (**COM**): znázorňuje modulární strukturu každého úkolu.
-

# Yourdon Structured Method YSM II

---

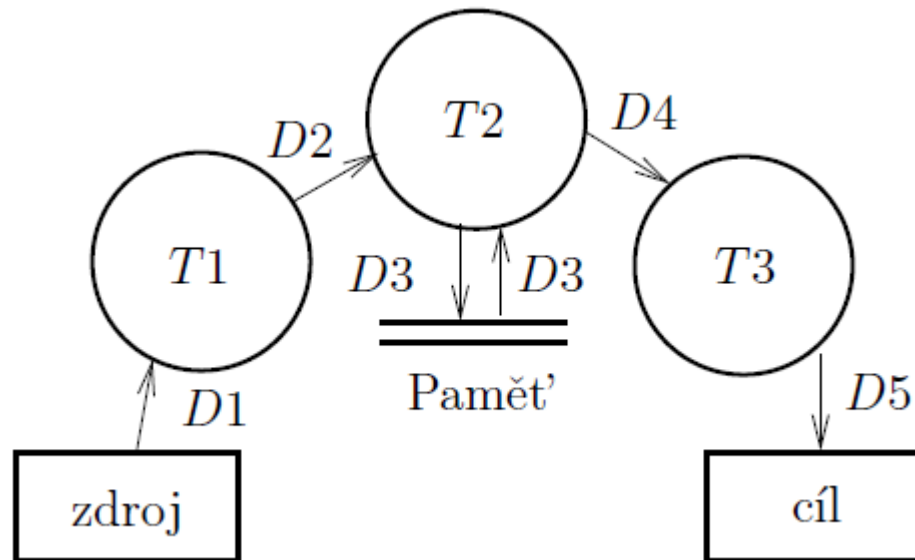
- YSM pokrývá : Analýzu požadavků, Specifikaci systému, Konstrukci systému, Implementaci
  - Vytváří 4 nezávislé modely
    - Datový : statický pohled na realitu, nejčastěji se používá ERA
    - Funkční: diagramy struktury funkcí, diagramy datových toků DFD a slovní popisy funkcí
    - Model řízení: Diagram stavů a přechodů (State transition diagram) a řídicí toky v DFD – popis jak se mají k sobě chovat funkce systému
    - Model struktury programového systému (patří již do System design)
  - Integrace z pohledu informací je zajištěna pomocí Data Dictionary DD
-

# Jacksonovy diagramy

Informace, které procházejí systémem, jsou podrobovány řadě transformací. Tyto transformace vyjádříme diagramem, ve kterém se obvykle používají následující značky:

- Obdélník vyznačuje vnější objekt, který představuje buď zdroj, odkud přicházejí informace do systému, nebo příjemce informací od systému. Od jednoho zdroje může vycházet i více datových toků.
- Kruh („bublina“) označuje proces, tedy transformaci dat.
- Spojnice označují přenosy informací mezi procesy navzájem nebo mezi procesy a vnějšími objekty. Šipky ukazují směr přenosu dat. Každá spojnice by měla být pojmenována.
- Dvojice vodorovných rovnoběžek označuje zařízení na ukládání dat.

Hierarchické  
stromové  
struktury



# Strukturovaná dekompozice v projektu IS

---

- V projektu IS se uplatňují zejména 2 hlediska:
    - Dekompozice funkční
      - Ve fázi plánování jde o stanovení základních nových funkcí IS a jejich vazeb , detailní plán vzniká na základě konceptu řešení
    - Dekompozice předmětová
      - Jde v podstatě o stanovení prvků HW a infrastruktury, které se projekt týká
  - Požadavek soudržnosti a jednoduchosti
    - Nesmí se narušit celistvost systému,
    - Dekompozice končí u samostatně řešitelných úloh
-

# Strukturované metodologie - SSADM

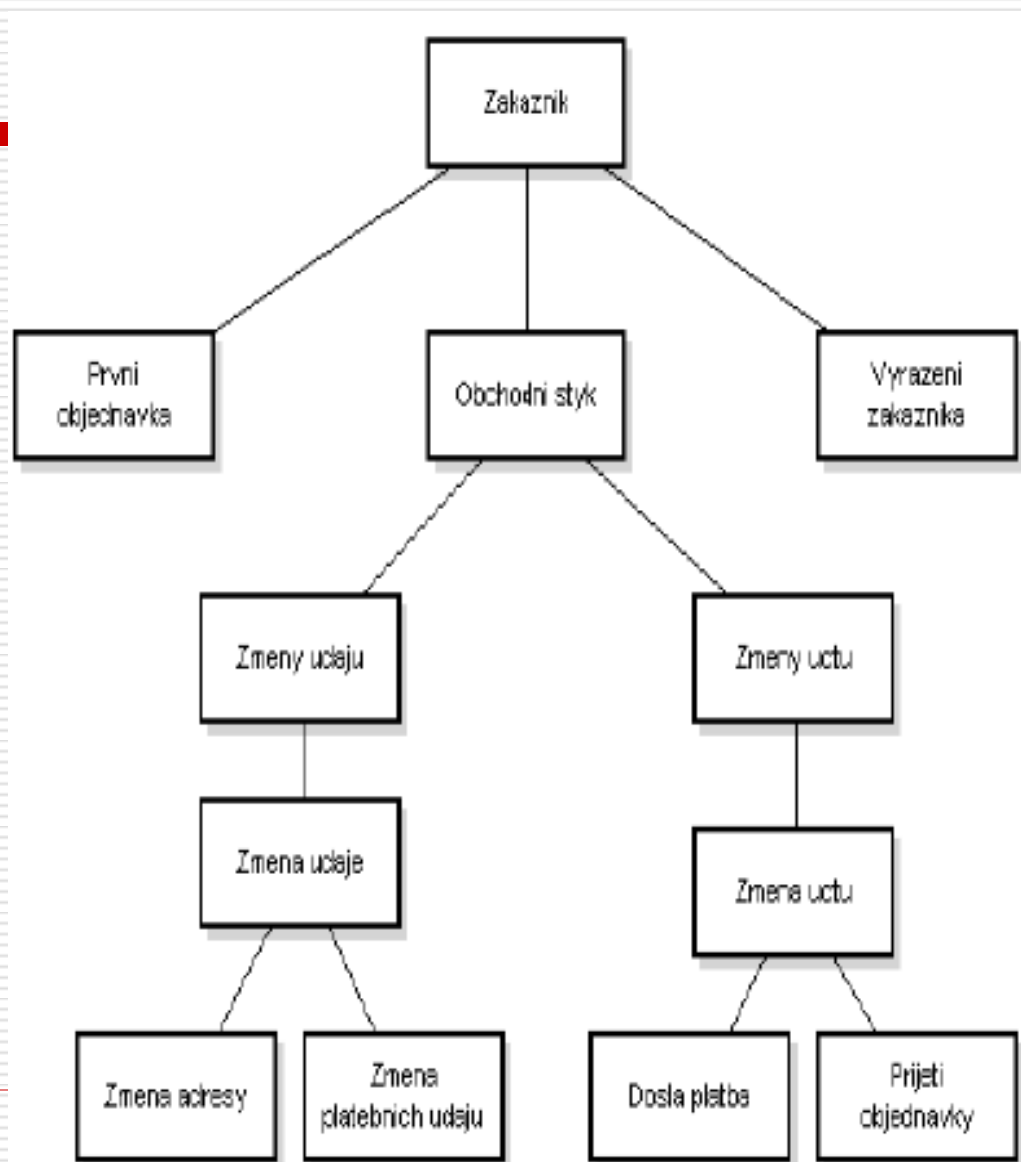
---

- SSADM – typický představitel strukturovaných metodologií
  - Používala se jako standard pro vládní projekty ve Velké Británii
  - SSADM (Structured Systems Analysis and Design Metology)
    - Analýza
    - Specifikace uživatelských a systémových požadavků
    - Výběr technických možností
    - Návrh struktury dat a procesů
    - Fyzický návrh
    - Nástroje SSADM se u různých autorů doporučení liší.
  - SSADM používá 3 pohledy na DATA
    - Logické datové struktury - informace a jejich vazby
    - Diagramy datových toků
    - Životní cykly entit
-

# Příklad životního cyklu entity

## Zákazník

SSADM

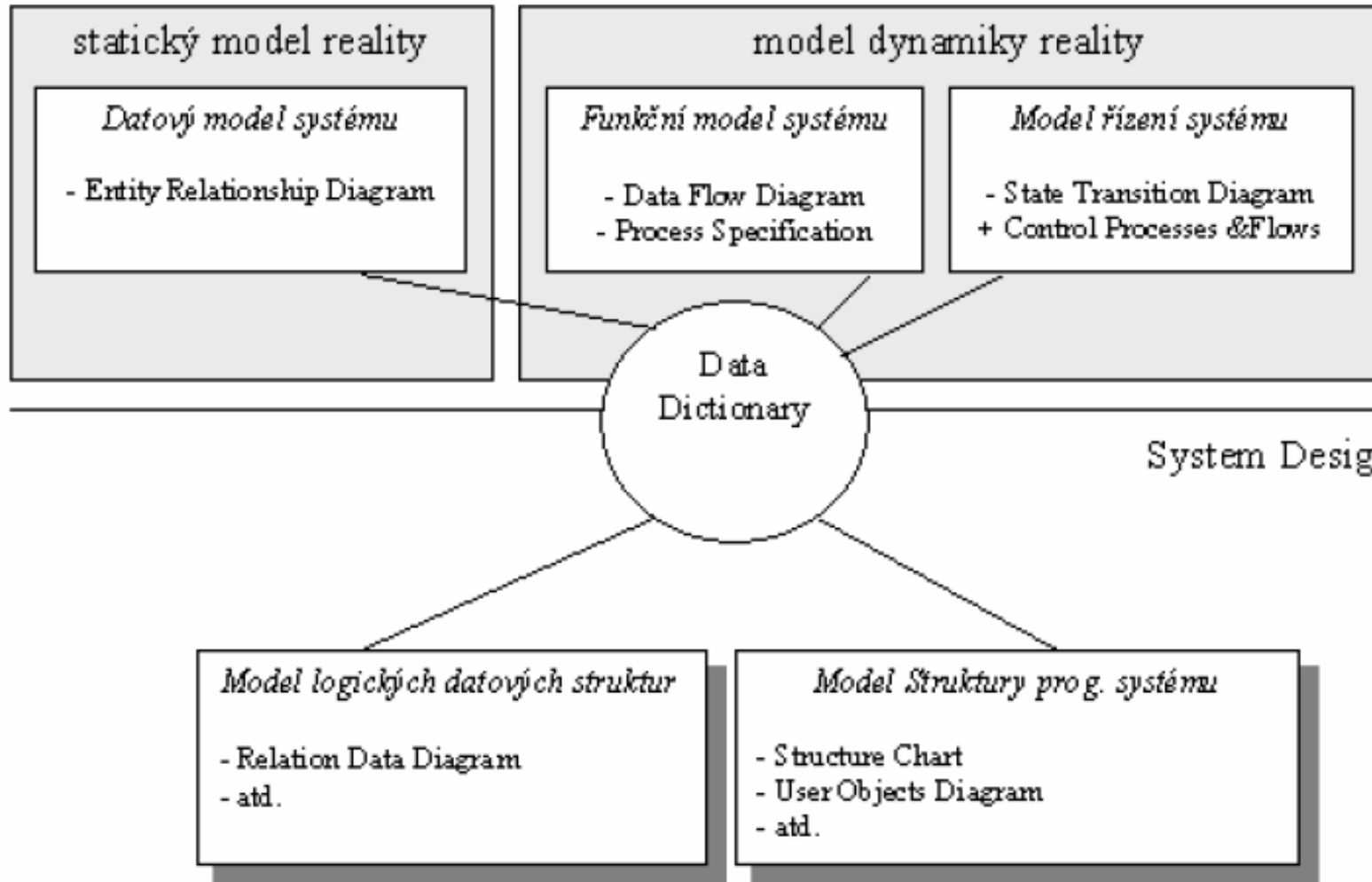




Na konceptuální úrovni je model reality tvořen třemi modely:

- datovým modelem (statický popis reality, popis prvků a vazeb, jimiž je realita tvořena),
- funkčním modelem (popisuje procesy a vztahy mezi nimi),
- modelem řízení (popisuje časové následnosti procesů).

Funkční model a model řízení tvoří popis dynamiky reality, popisují, jakým způsobem se realita mění.



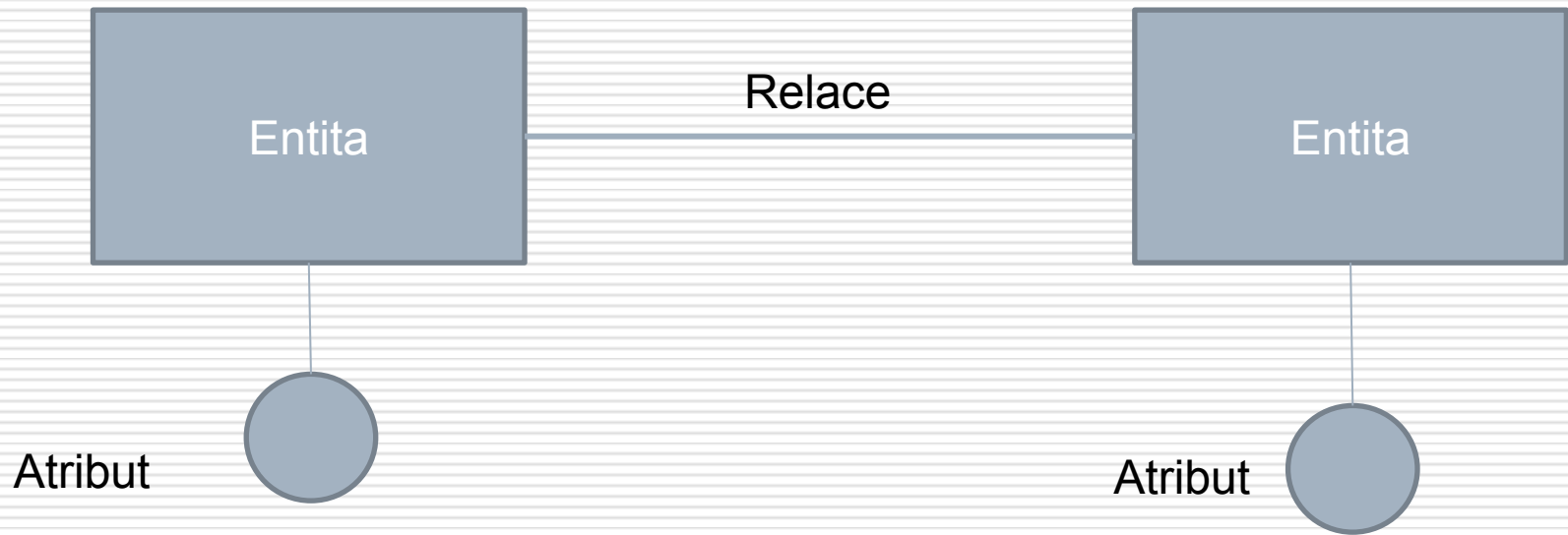
# Datový model ERA (Chen)

---

- Objekty, vztahy mezi objekty a vlastnosti objektů (vztahů)
  - Entita – předmět našeho zájmu.
    - Typ – student
    - Výskyt - student OXiiiiiii
    - Má definici, popis, jak vzniká a zaniká
  - Vztah – důležité vztahy mezi entitami
    - Násobnost - binární až n-ární (kolik entit je vztahem spojeno)  
např. ředitel-řídí-podnik je binární ale ředitel, náměstkovo-vedení podniku je n-ární vztah
    - Kardinalita – má vazbu na výskyty 1:1, 1:n, n:m (1:1 na každé straně je jen jeden výskyt např. děkan-řídí-fakulta)
    - Odvoditelnost – hledáme ty vztahy, které se nedají odvodit z jiných vztahů
    - Parcialita – vztah totální (musí existovat vždy), parciální
-

# ERA

---



# Chenův model ERA II

---

- Atributy
    - jsou podrobnosti (vlastnosti) objektů nebo vztahů, které v modelu zkoumáme
    - Atributy
      - Identifikační ( např. rodné číslo)
      - Parciální – nemusí mít hodnotu (např. č. pasu)
      - Opakované (např. jazykové vlastnosti)
  - Konceptuální schéma
    - Textová část (podrobné verbální popisy případně s logickými vztahy)
    - Grafická část (všechny entity, vztahy mezi entitami, identifikační atributy)
    - Kvantifikace (počty výskytů, počty opakování, frekvence přístupů, minimální, maximální a průměrné počty)
-

# ERA – definice jinak

---

## □ Entita

- Obecná
- Silná/kmenová/základní (nezávislá na jiných entitách)
- Slabá / popisná (existenčně závislá na jiných)
- Vazební / asociativní (realizuje vazbu)

## □ Vztah

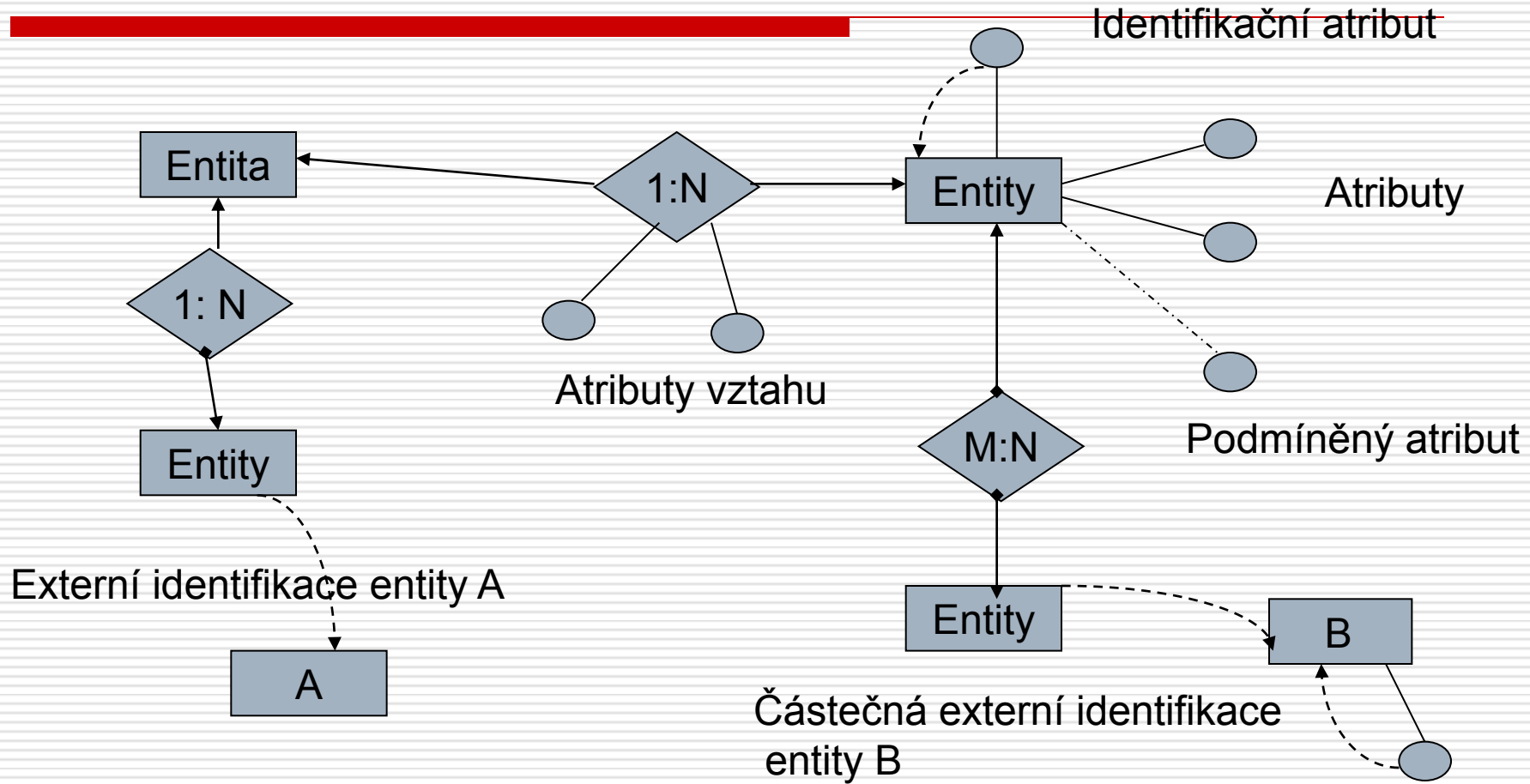
- N-ární / polyární
  - Binární
  - Kardinalita (max a min počet výskytů v určitém vztahu 1:1, 1:N, M:N)
  - Volitelnost / parcialita (vztah se nemusí týkat všech výskytů entity např. 0:N)
  - Výlučnost
  - Externí id / slabý vztah (nestačí vlastní atributy, je nutná externí identifikace)
-

# ERA – definice jinak II

---

- Atribut
    - Jednoduchý
    - Složený
    - Základní
    - Odvoditelný
  - Klíč
    - Primární
    - Cizí ( je to klíč, který je primárním v jiné entitě, slouží pro vyjádření vztahů v datovém modelu na technologické nebo implementační úrovni)
    - Alternativní
    - Sekundární / nejednoznačný (atributy důležité pro přístup, další třídění atd.)
-

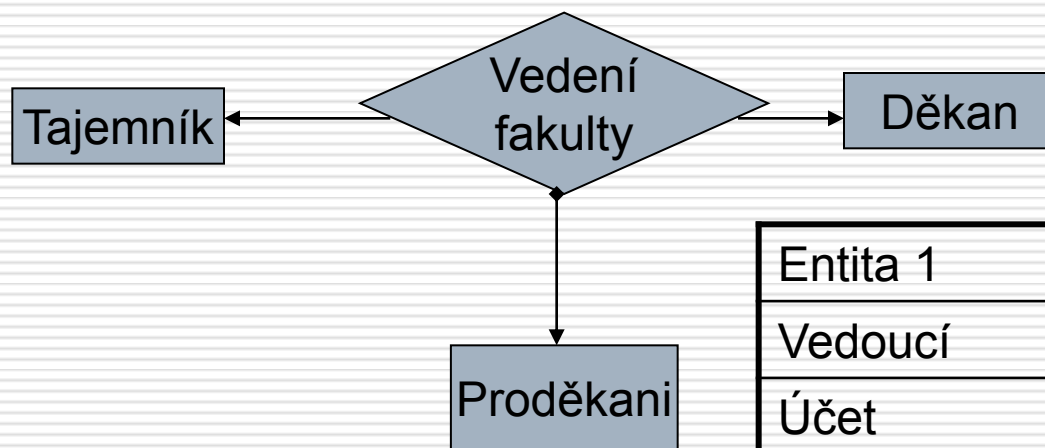
# Entity





# Násobnost a kardinalita – např.

---



Entita 1	Vztah 1: n	Entita 2
Vedoucí	řídí	Pracovník
Účet	skládá	Položka
Vlak	veze	Vůz
Lékař	léčí	Pacient

---

# Normální formy

---

- ❑ 1. normální forma – atributy obsahují pouze atomické hodnoty. ( příklad: 1 osoba a 2 tel. čísla) – rozdělit.
- ❑ 2. normální forma – každý neklíčový atribut je závislý na celém primárním klíči
- ❑ 3. normální forma – všechny neklíčové atributy jsou vzájemně nezávislé

# Postup přípravy ER diagramu

---

- Výběr hlavních objektů (entit)
- Definice vztahů mezi entitami (včetně kardinalit)
- Přidání atributů entitám (zejména identifikátory)
- Definice hierarchie (hledají se vztahy mezi generalizací a specializací )
- Odstranění tranzitivních vztahů (těch, které se dají odvodit z jiných)
- Odstranění nadbytečných entit
- Ověření úplnosti
- Výsledek – konceptuální schéma datové základny

# Implementace datové základny

---

- Konceptuální schéma nebere do úvahy, v jakém prostředí má být systém zaveden.
  - Proto jsou nutné následující kroky:
    - Volba prostředí (databázový SW, HW, ...)
    - Tvorba logické struktury datové základny
    - Vytvoření fyzické struktury datové základny
  - Důležité otázky pro implementaci:
    - Způsob práce s daty (client-server , on line, dávka, řízeno událostmi...)
    - Přístupové metody a frekvence ( klíče, sekvenční, metody hledání)
    - Počty záznamů každého typu
    - Současné přístupy a očekávané doby odezvy,
    - Požadavky na bezpečnost a omezení uživatelů
    - Odvoditelné atribut (počítaná pole) a jejich podíl
    - Nutné kompromisy v čistotě návrhu (duplicitní tabulky, pole...)
-

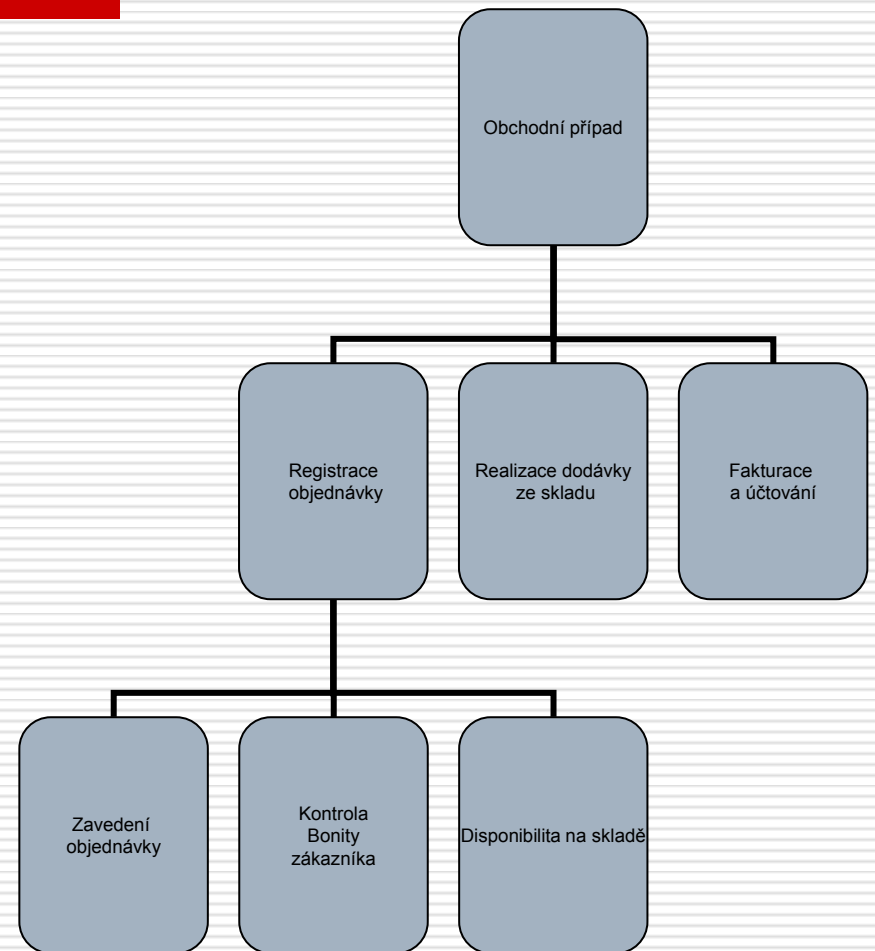
# Logická a fyzická struktura dat

---

- Logická struktura
    - Je implementací konceptuálního modelu
    - Abstrakce vztahů mezi daty
      - Lineární, stromová, relační struktura
      - Výskyty, četnost vztahů,
      - Zde též bereme do úvahy potřeby na HW
  - Fyzická struktura
    - Zavedení reálných (testovacích ) dat do struktur
    - Test splnění požadavků uživatele na data a vlastnosti jejich poskytování
    - Úpravy fyzických dat, případně změny v logických strukturách
-

# Funkční analýza FSD

- ❑ Top Down přístup k hierarchii funkcí
- ❑ Dynamické hledisko (posloupnost funkcí a podmínky řízení jejich pořadí) – zpravidla text nebo tabulka

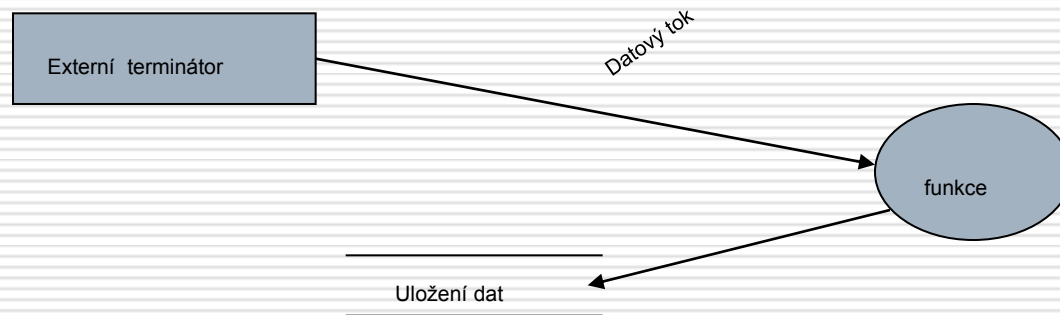


Jacksonův SD

# Diagram toku dat DFD

---

- Obsahuje:
  - Datové prvky
  - Funkce (procesy, transformace)
  - Data store – systém uchovávání dat
  - Terminátory – prvky okolí, které jsou zdrojem, nebo cílem datových toků
  - Spojení DFD a top down principu se často používá u velkých návrhů



# State transition diagram STD

Model řízení je strukturovanými metodami zpravidla vyžadován pouze pro případ Real-time systémů, kde jsou časové následnosti procesů často jedinou podstatnou skutečností.

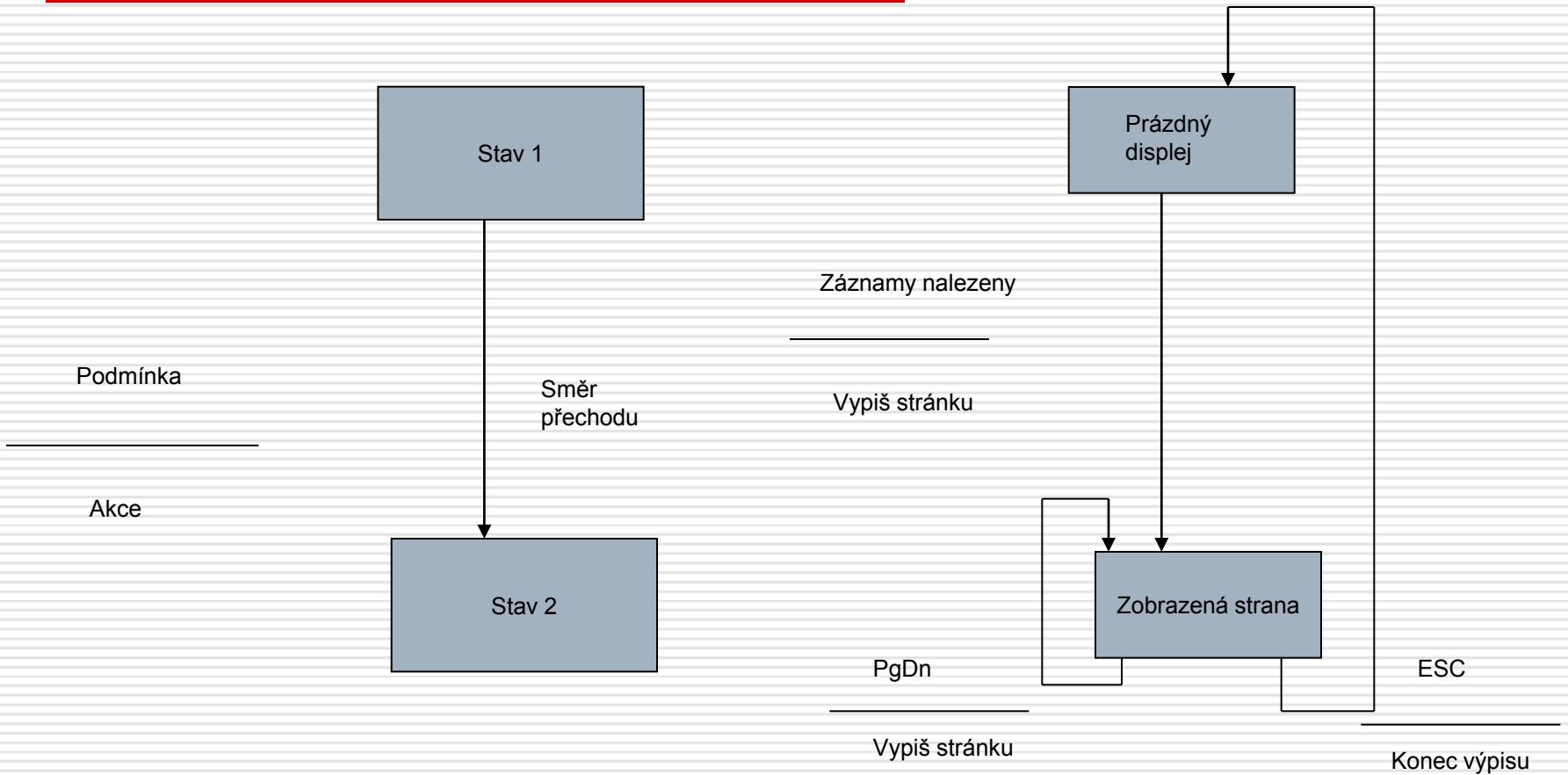
K modelování chování systému, který závisí na čase slouží STD. Diagram stavů a přechodů je nástrojem popisu struktury řídicího procesu.

STD popisuje:

- *stavy řízeného systému* (znázorněny obdélníky) – každý stav je výchozím bodem pro případně přechody do dalších stavů,
- *možné přechody mezi stavy* (znázorněny šipkami),
- *důvody (podmínky nebo signály)* jednotlivých přechodů,
- *způsoby (akce vedoucí k uskutečnění)* jednotlivých přechodů.



# Diagram přechodů a stavů STD



# Vztahy mezi nástroji I

---

- DFD a DD
  - Každý datový tok a data store musí být definován v DD (vztah k ontologii)
- Specifikace procesu a DFD + DD
  - Každý odkaz na data ve specifikaci procesů k DFD: Musí použít název dle DD
    - Nebo mít název lokálních dat dle DD
- Vztahy DFD ke specifikaci procesů
  - Každý proces který už není rozepsán na nižší úroveň DFD musí být popsán ve specifikaci procesů
  - Každý specifikovaný proces musí být obsažen v některém DFD nejnižší úrovně
  - Každému výstupnímu toku z procesu musí odpovídat WRITE a každému vstupnímu zase READ

# Vztahy mezi nástroji II

---

- Vztahy DD+DFD ke specifikaci procesů
    - Každý datový prvek v DD musí být použit ve specifikaci procesů, nebo DFD případně při popisu jiného datového prvku
  - Vztahy ER diagramu + DFD ke specifikaci procesů
    - Každý Data store v DFD musí být v ERD zastoupen jako objekt nebo vztah nebo kombinace obojího
    - Datové prvky v DD popisují jak data v ERD tak data v DFD, to znamená že data v DD musí být v DFD i ER diagramu
    - Specifikace procesů musí obsahovat operace CREATE a DELETE pro každý objekt a vztah uvedený v ER diagramu
    - Atributy každého objektu musí být nastaveny některým procesem v DFD
  - Vztahy mezi DFD a STD
    - Každý řídicí proces v DFD musí mít svůj STD
    - Každá podmínka v STD odpovídá vstupnímu řídicímu toku v DFD a naopak
    - Každá akce v STD odpovídá výstupnímu řídicímu toku v DFD a naopak
-

# Objektové přístupy k zobrazení reality – základní charakteristika

---

- Princip je ve spojení dat a služeb
  - Metodologie a metody
    - Yourdon/Coad OOA/OOD (Yourdon&Coad Prentice Hall 1990)
    - Object Modelling Technique OMT (James Rumbaugh „Object oriented Modelling and Design Prentice-Hall 1991) viz dále Rational Rose a Select
  - Nástroj např. UML
  - Objektové metody však nenahrazují plně strukturované přístupy , stále jsou důležité diagramy procesních a datových toků
-

# Objektově orientované metodologie

---

## □ OOA/OOD

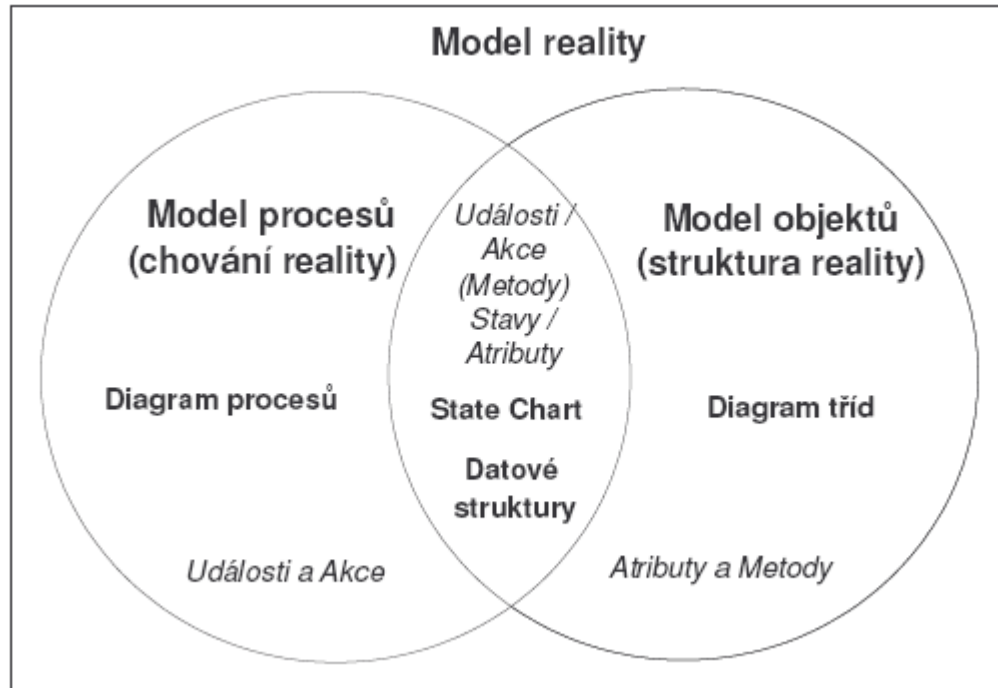
- Analýza : 5 vrstev – subjekty (problémové oblasti), objekty, struktury, atributy, služby
- Design: definuje třídy v problémové oblasti, třídy lidské interakce, třídy správy systému, třídy správy dat (přístupu k databázím)

## □ OMT (Object Modeling Technique)

- Fáze vývoje systému : analýza, systémový design, objektový design, implementace a testování
  - Objektový model: definice tříd a jejich vztahů, atributů a metod  
Statická struktura systému
  - Dynamický model: změny stavů objektů – stavové diagramy STD, mapa událostí, diagram událostí (reakce systému na vstupy)
  - Funkční model : popisuje co systém dělá, ne jak to dělá – obdoba DFD
- 
- Model jednání - obdoba Use case – specifikace a určení hranic systému

# Souvislosti modelů

---



# Objektové technologie

---

- Základním pojmem objektově orientované technologie je objekt.
  - Základní myšlenka objektového přístupu spočívá v tom, objekt zahrnuje také činnosti, které jsou s objektem svázány. Spojení datových struktur s algoritmy nazýváme zapouzdření (angl. ENCAPSULATION) a činnosti zapouzdřené do objektu označujeme jako metody (angl. METHODS).
  - Objekty se společnými vlastnostmi tvoří tzv. třídy (angl. CLASSES).
  - Konkrétní výskyt určitého druhu objektu se nazývá jeho instance

# Objektové metody – definice I

---

- Třída
    - Zobecnění reálných objektů
    - Popisná charakteristika : atribut (omezení)
    - Abstraktní – nemá instance objektů
  - Metoda
    - Popisuje chování objektů dané třídy
    - Popisná charakteristika: příznak
  - Závislost
    - Pokud jedna třídy využívá jinou třídu např.metoda „zobraz menu“ u jedné třídy volá objekty z třídy „menu“
  - Rozhraní
    - Skupina operací určující chování třídy a její vztah k jiným třídám
    - Vztah mezi třídou a rozhraním : realizace
-



# Objektové metody – definice II

---

- Viditelnost (zapouzdření)
    - Veřejná, private, protected
  - Dědičnost
    - Každý objekt dědí atributy a metody třídy do které patří i její nadtřídy, pokud existuje
    - Jeden rodič – jednoduchá dědičnost
    - Žádný rodič – základní třída
    - Žádný potomek – listová třída
  - Asociace – vzájemný vztah objektů
    - Jednosměrné i obousměrné
    - ROLE – každá třída má v asociaci roli
    - LINK – vazba – instance asociace
    - AGREGACE – objekt je agregací objektů jiných tříd
-

# Objektové technologie II

---

- Důsledek pojmu Třída: dědičnost
  - Nový objekt určité třídy dědí všechny vlastnosti této třídy. Hovoříme o rodičovském objektu , o odvozeném objektu, který dědí ( INHERITS) všechny atributy a metody svého předka. Potomek však může být rodičem pro další objekt.
  - Polymorfismus – určitou vlastnost, metodu sdílí celá hierarchie ale lze ji na určité úrovni upravit, přizpůsobit.

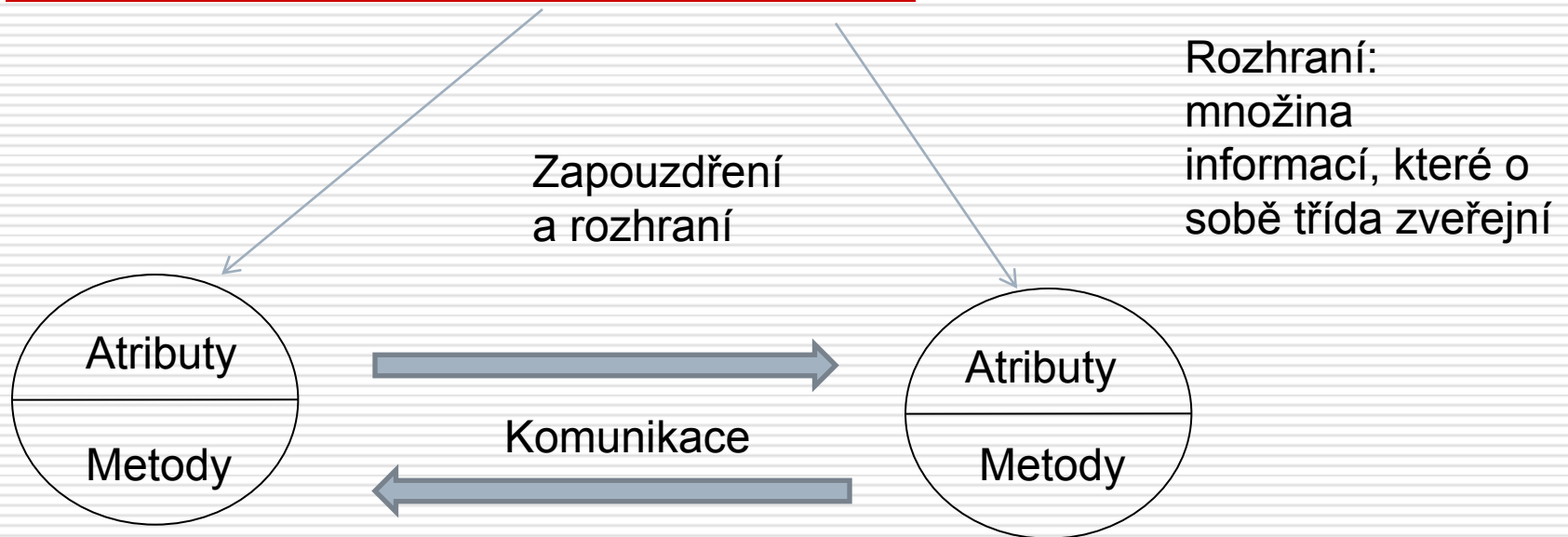
# Zapouzdřenost

---

- Striktně rozlišujeme vnějšek a vnitřek třídy
  - Vnitřní atributy, metody a rozhraní nejsou z vnějšku viditelné
  - Vnějškem třídy se rozumí komunikace mezi objekty
- Základní výhody objektových metod. Proč?
  - Máme-li příliš složitý problém, snažíme se jej rozložit
  - Rozkládáme tedy problém na nižší objekty, třídy
- Jaký je zde rozdíl oproti strukturované metodě?
  - Strukturované metody – jednotlivé dekompozice dle pohledů.
  - Objektově – celek včetně metod, atributů a dědičností

# Princip rozhraní se zapouzdřením

---



```
Public Interface Moje_Rozhraní { }
```

```
...
```

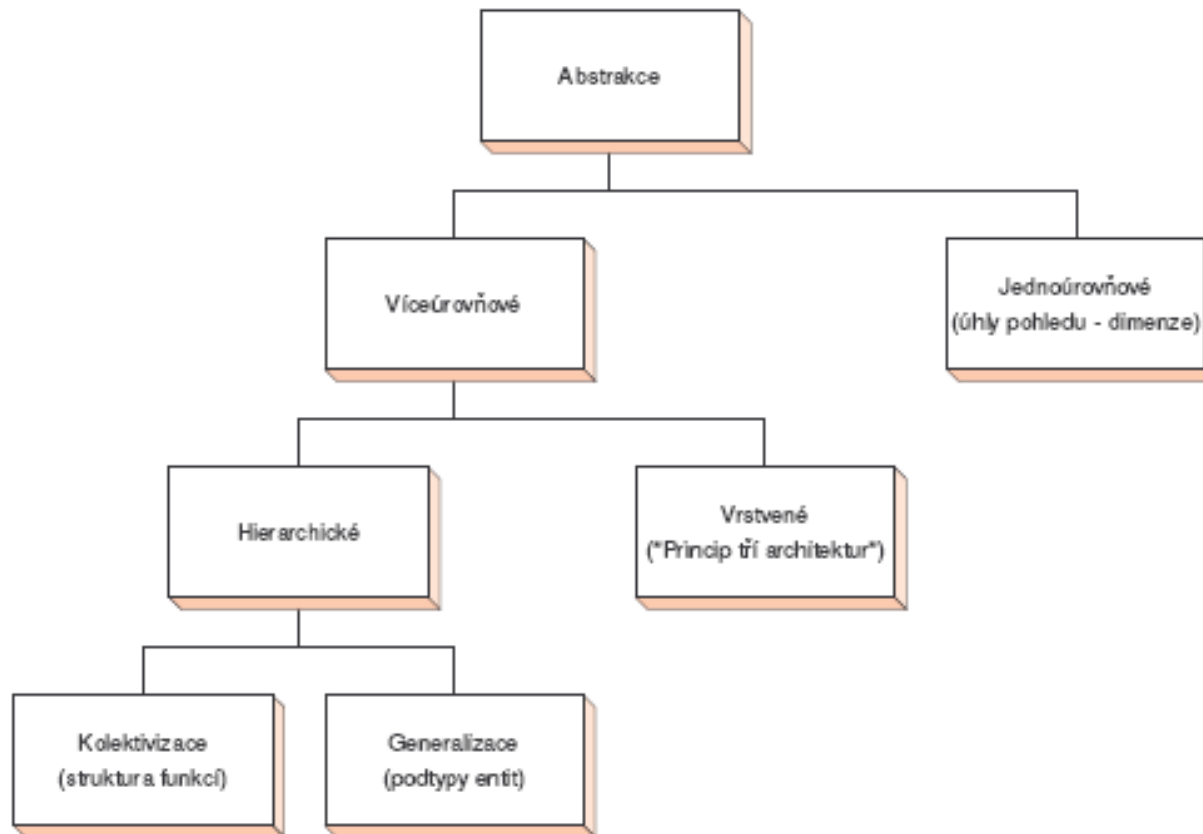
```
Public Class Moje_Třída implements Moje_Rozhraní { }
```

# Postup

---

- ❑ Provede se dekompozice až do tříd
- ❑ Definují se třídy a jejich rozhraní
- ❑ Pak se jde dovnitř tříd
- ❑ Když se ukáže, že je třeba rozhraní dodefinovat, provede se iterace s případnou úpravou tříd a rozhraní
- ❑ Poznámka:
  - Třídy v různých metodikách mohou být ekvivaletní objektům v UML.

# Abstrakce



# Generalizace a specializace

---

## □ Generalizace

- Objekt vyšší úrovně je nositelem společných vlastností
- Tento objekt je nadtypem svého podtypu
- Jednotlivé podtypy jsou navzájem disjunktní

## □ Specializace

- Představuje zjemnění třídy do podtřídy

# Záměna výskytů za jejich abstrakce

---

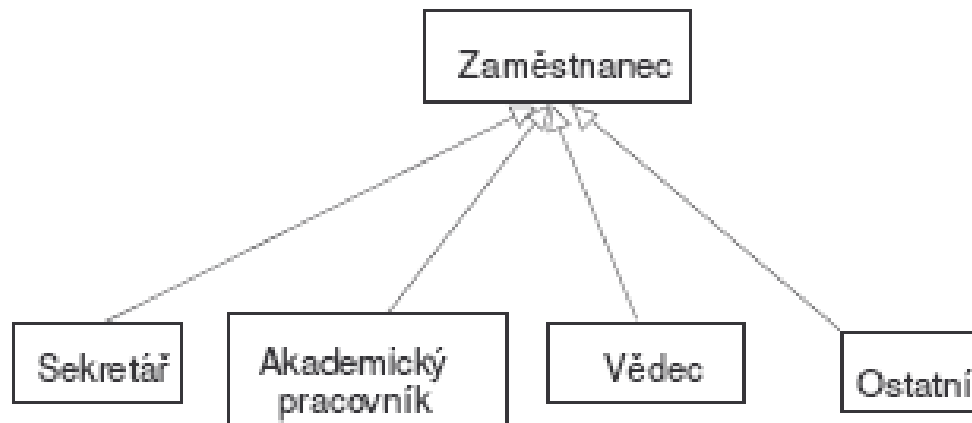
- Zobecněním – generalizací provádím abstrakci a definici vyšší úrovně
- Příklad:
  - Máme nastavená pravidla jak tvořit obchodní zakázky
  - Každá zakázka tím získává určité atributy
  - Každá zakázka je zpracovávána v podstatě stejným způsobem
  - Zobecnění pro všechny zakázky daného typu – generalizace
  - Výsledek : třída zakázek definující pravidla týkající se výskytů všech zakázek daného typu.



# Příklad generalizace

---

Generalizace v datovém/objektovém modelu



# Jiný příklad

---

## Člověk

### ■ Skládá se

Kostry

Svalů

Vnitřních orgánů atd.

### ■ Avšak: tato statická definice nic neříká o funkcích jednotlivých orgánů

■ Srdce pohání krev

■ Krev se okysličuje v plicích

■ Atd.

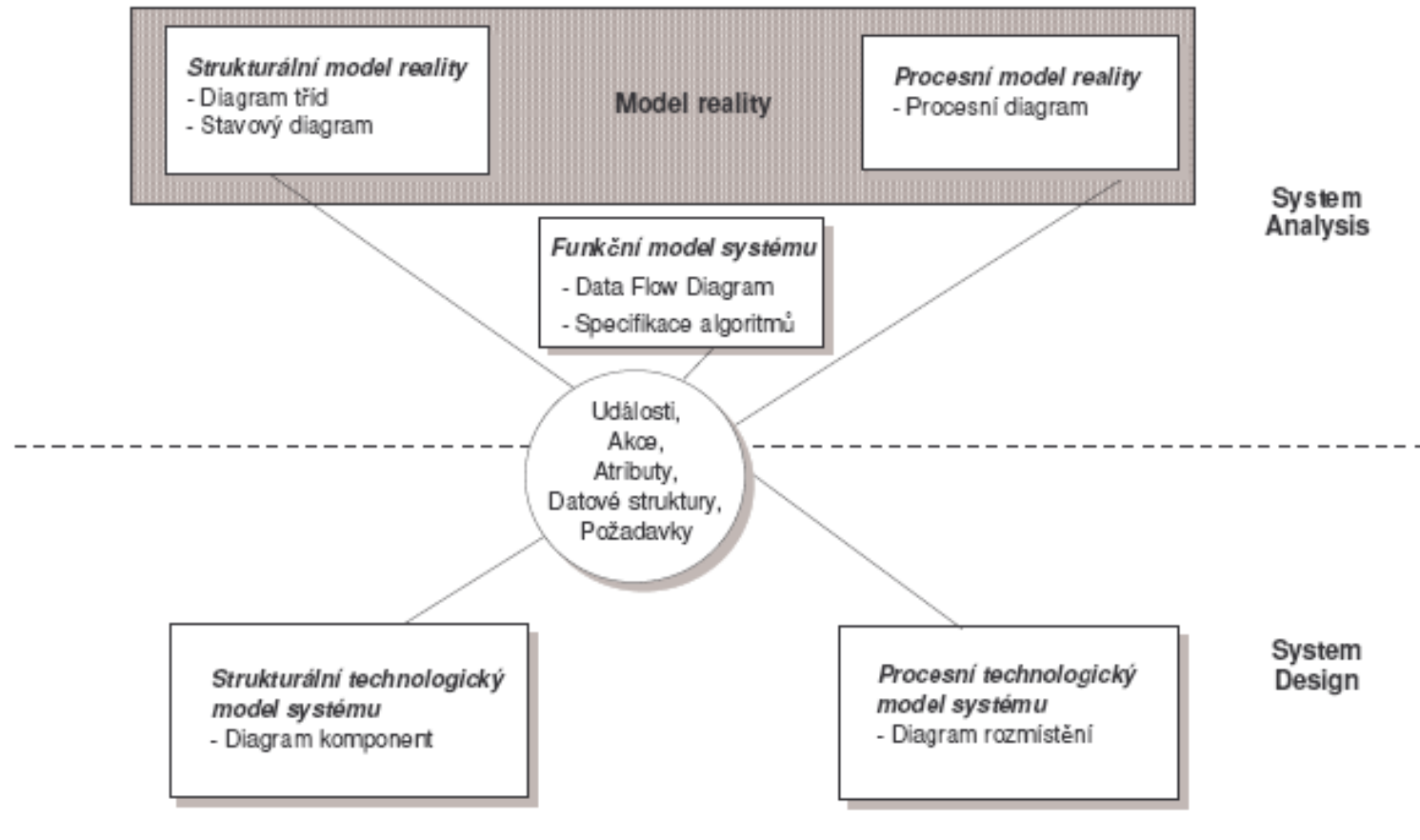
## Ještě vyšší úroveň generalizace: třída savců atd.

# Modely používané v OO metodikách

---

- Model tříd a stavový model
  - Popisuje objektový pohled na realitu
- Procesní model
  - Klíčové procesy a jejich produkty
- Funkční model
  - Klíčové funkce
- Struktura technologie
- Procesně-technologický
  - Umístění logických komponent na fyzických komponentách

# Modely při objektovém návrhu IS



Popis současného stavu	Organizační struktura zákazníka		
	Technické vybavení		
	Stávající programové vybavení	Operační systémy Aplikační vybavení	
Návrh řešení	Datová základna		
	Počítačová síť		
	Výpočetní technika	Servery	
		Pracovní stanice	
		Tiskárny	
	Systémové programové vybavení	Servery Pracovní stanice	
	Zálohování dat		
	Standardní typový programový balík		
	Aplikace a dodatečných modulů		
	Moduly vytvořené na zakázku		
	Návrh organizace datové základny		
	Zajištění komunikace mimo centrálu	Komunikace s jinými IS Komunikace se vzdálenými pracovišti zákazníka	
	Nutná organizační opatření vztahující se k zavedení IS		
	Návrh konverze dat		
	Návrh školení		
Organizace akceptačních zkoušek			
Rízení projektu			
Nastavení a řešení požadovaných funkcí	Finance	Oběh účetních dokladů	
		Účtová osnova atd.	
	Prodej a pohledávky	Správa zákazníků	
		Zakázky a prodej	
		Centry atd.	
	CRM/ Servis	Předměty servisu	
		Servisní zakázky Dispečink atd.	
	Nákup a závazky	Správa dodavatelů Objednávky a nákup atd.	
	Zásoby	Zboží	
		Evidence skladových položek atd.	
	Specifické funkce	Komunikace s partnerskými firmami	
Propojení s B2B atd.			
Zakázkové moduly	Rozšířený systém uživatelských oprávnění		
	Rezervace zboží na servisní zakázku		
	Specifické sestavy atd.		

Zdroj: vlastní zpracování



Děkuji za pozornost.

Otázky?