

# Diagram datových toků Data Flow Diagram (DFD)

Václav Řepa

KIT VŠE Praha

[repa@vse.cz](mailto:repa@vse.cz)

[www.panrepa.org](http://www.panrepa.org)

# Principy přístupu k vývoji IS kam DFD patří

# Základní principy vývoje IS organizace

## Princip modelování

- **Objektivním základem** implementace informačního systému musí být reálný svět: **reálná fakta, existující mimo organizaci (a nezávisle na ní)**

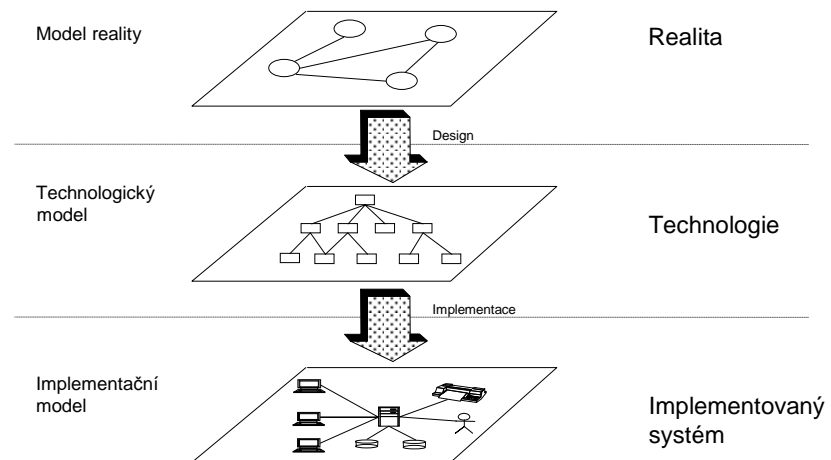
- Model **objektů** jako **souhrn atributů – kritických faktorů**
- Model **procesů** jako **souhrn reakcí na změny kritických faktorů (události)**

## Princip abstrakce

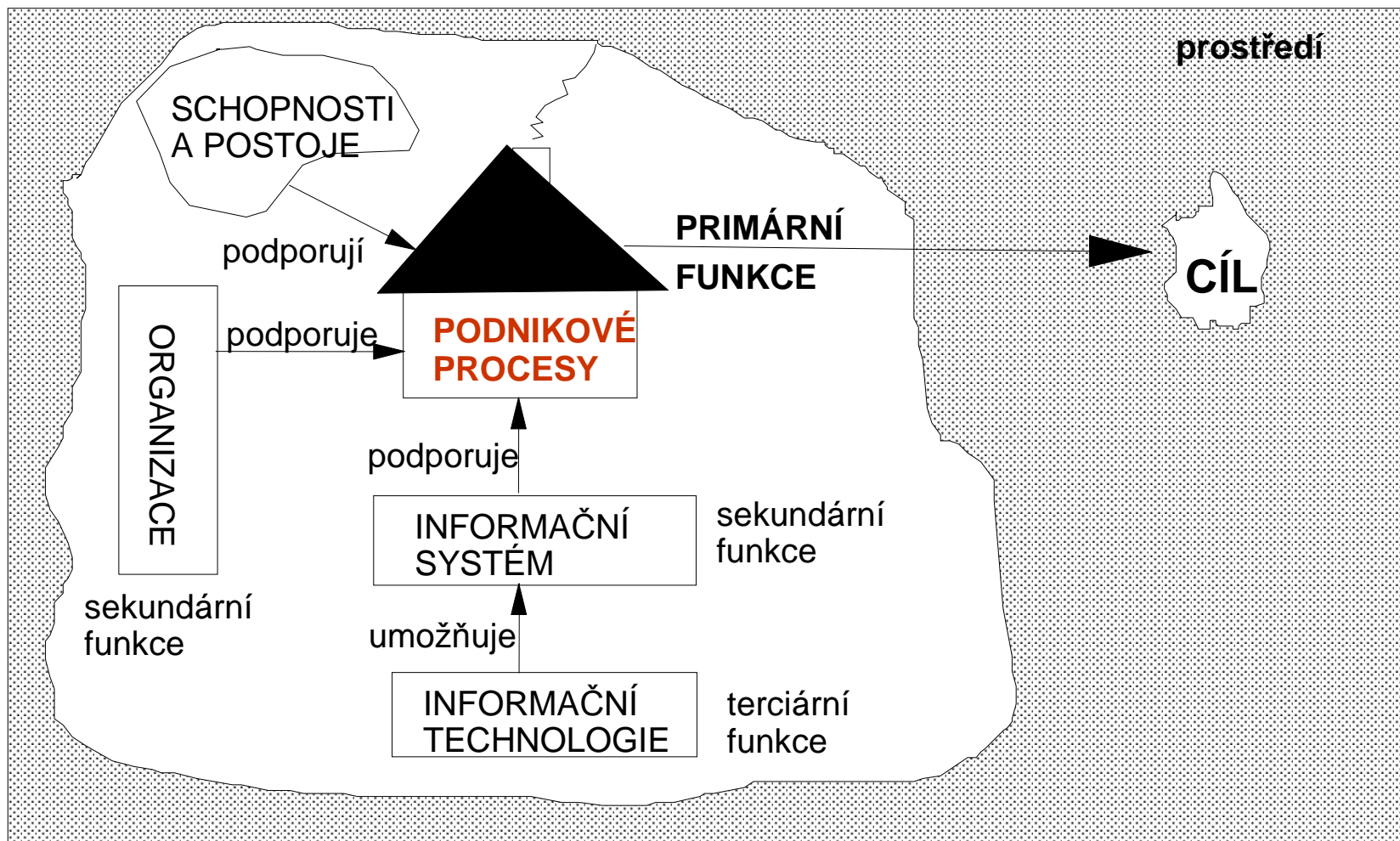
- veškerá podstatná fakta jsou analyzována do detailu a detaily abstrahovány do celků s použitím hierarchických abstrakcí:
  - Celek - část (proces - subprocess)
  - Typ – pod-typ (hierarchie tříd, dědičnost)

## Princip tří architektur

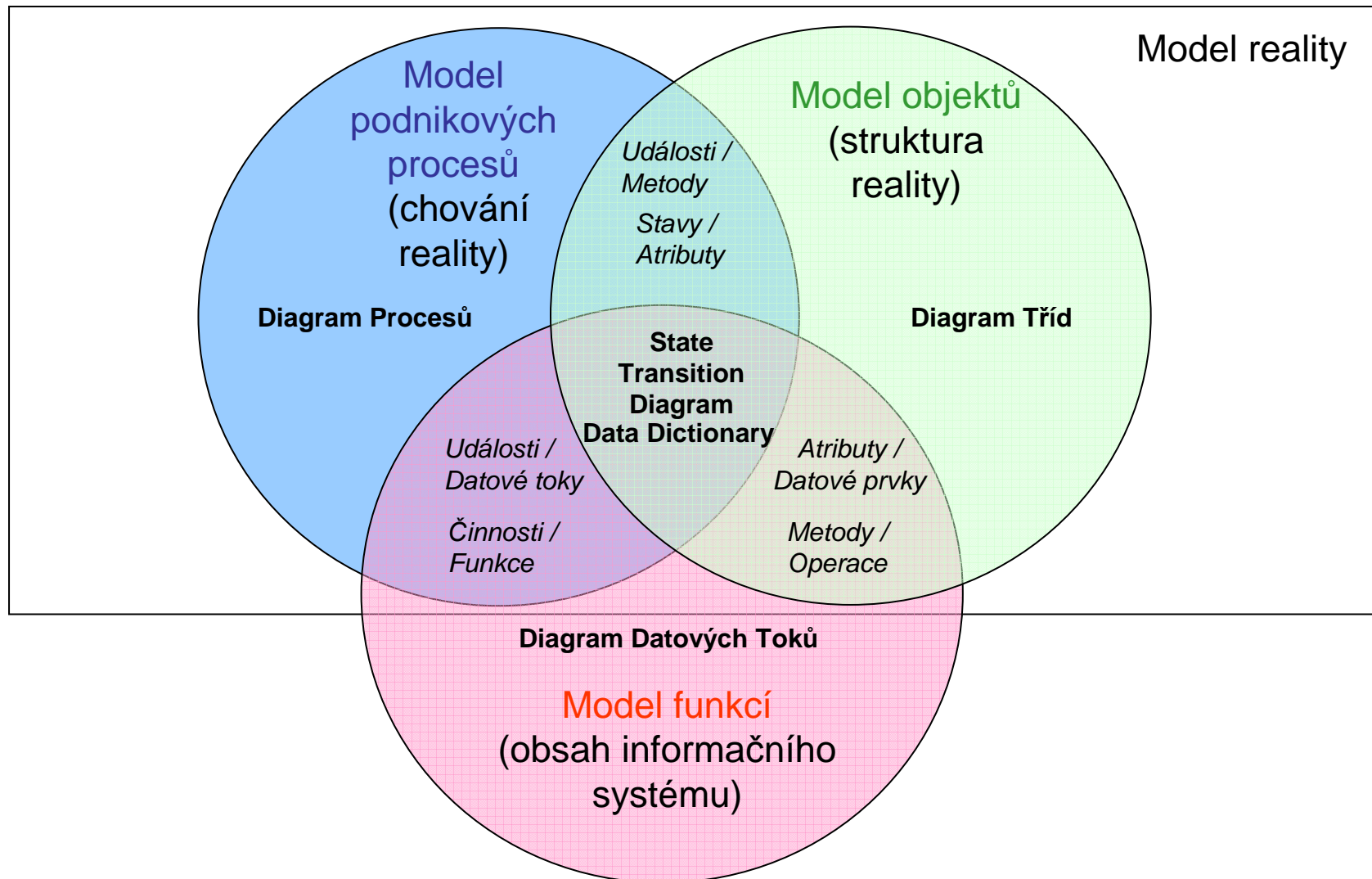
- potřeba rozlišovat:
  - přirozené vlastnosti objektů / procesů
  - vlastnosti objektů / procesů dané konkrétními podmínkami použité technologie a implementačního prostředí



# Základní podniková struktura a její infrastruktury

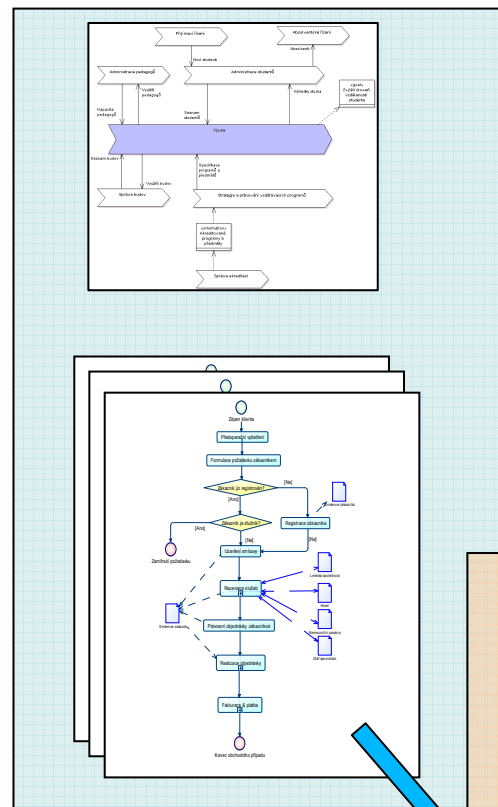


# IS jako model reality



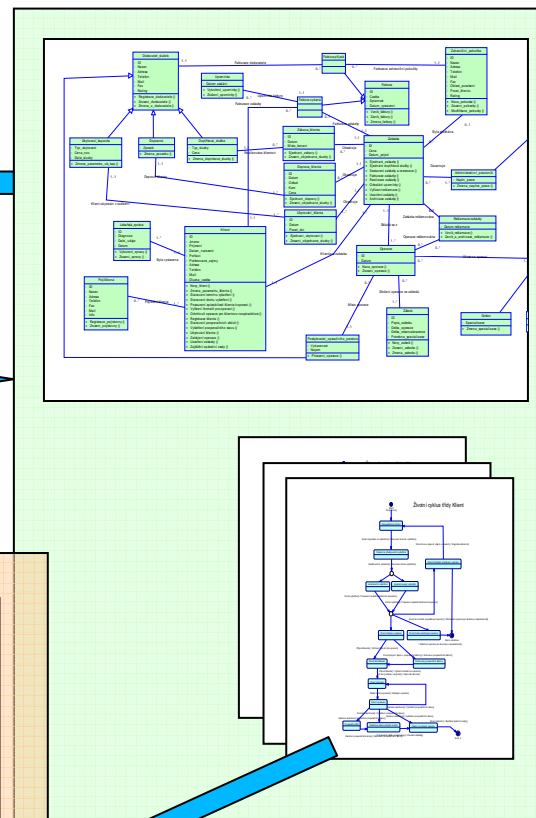
# Přehled analytických modelů

**Model podnikových procesů**  
(Globální model procesů, Process Diagrams)



Události a jejich  
účelové  
kombinace

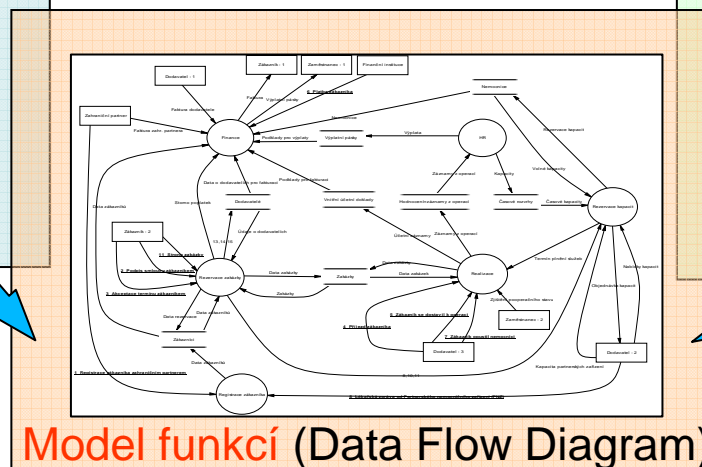
**Model objektů**  
(Class Diagram, State Charts)



Události a jejich  
obecný  
kontext

Produkty, vstupy, výstupy,  
aktéři, business omezení  
procesů (životní cykly  
objektů)

Účelové kombinace ŽC objektů,  
kontext chování objektů



# Co je Data Flow Diagram (modelování funkčnosti IS)

# Data Flow Diagram

- 70tá léta 20. stol.
  - První zmínky v metodice SADT (Marca/Mc.Gowan)
- 80tá léta – 90tá léta 20. stol.
  - Rozpracování DFD zejména v díle DeMarca
  - Ústřední role v metodice Strukturované analýzy a návrhu IS (Ed Yourdon),
  - Důležitý prvek metodiky OMT (Rumbaugh a kol.)
  - Klíčový diagram standardů IDEF (IDEF0 - Marca)
  - Základní podporovaný diagram v nástrojích CASE (Computer Aided System Engineering)
- 2. polovina 90tých let 20. stol.
  - Postupně mizí z metodik i nástrojů CASE, zjevně v souvislosti se vznikem UML (Unified Modelling Language)
  - Politická mrtvola
- 2. polovina 10tých let 21. stol.
  - Postupný vzestup zájmu a návrat do nástrojů CASE

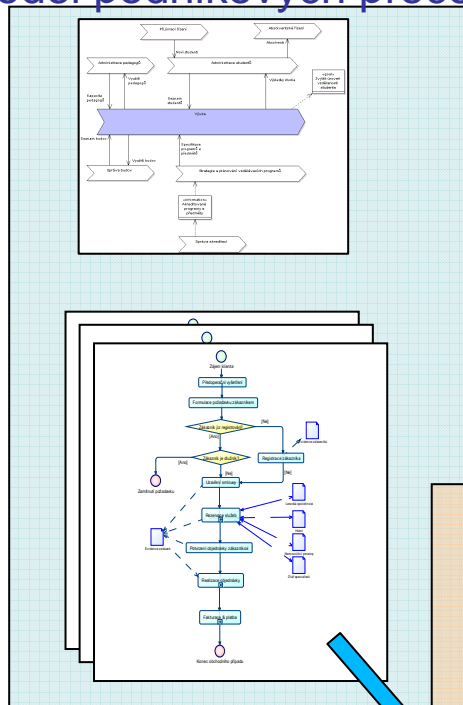


# Co je Data Flow Diagram (DFD)

- Použití v analytické fázi vývoje IS - modelování funkčnosti systému
  - cílem je popsat funkčnost informačního systému
  - funkčnost = potenciál chování systému
  - cílem chování informačního systému je odrážet dění v reálném (business) systému – viz Princip modelování
  - neboli modelovat relevantní kombinace událostí a akcí
    - ať obecně platné (neb nutno respektovat obecná „business rules“)
    - tak záměrné (neb nutno podporovat „business procesy“)
- Popisuje funkce a jejich vazby:
  - Datové toky
  - Datastory (úložiště dat)
- Notace E.Yourdon (DeMarco)

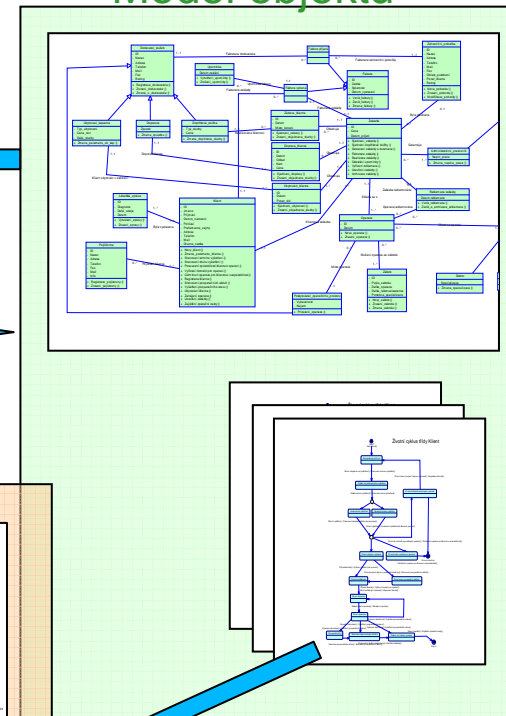
# Data Flow Diagram modeluje relevantní kombinace událostí a akcí

Model podnikových procesů



Události a jejich účelové kombinace

Model objektů



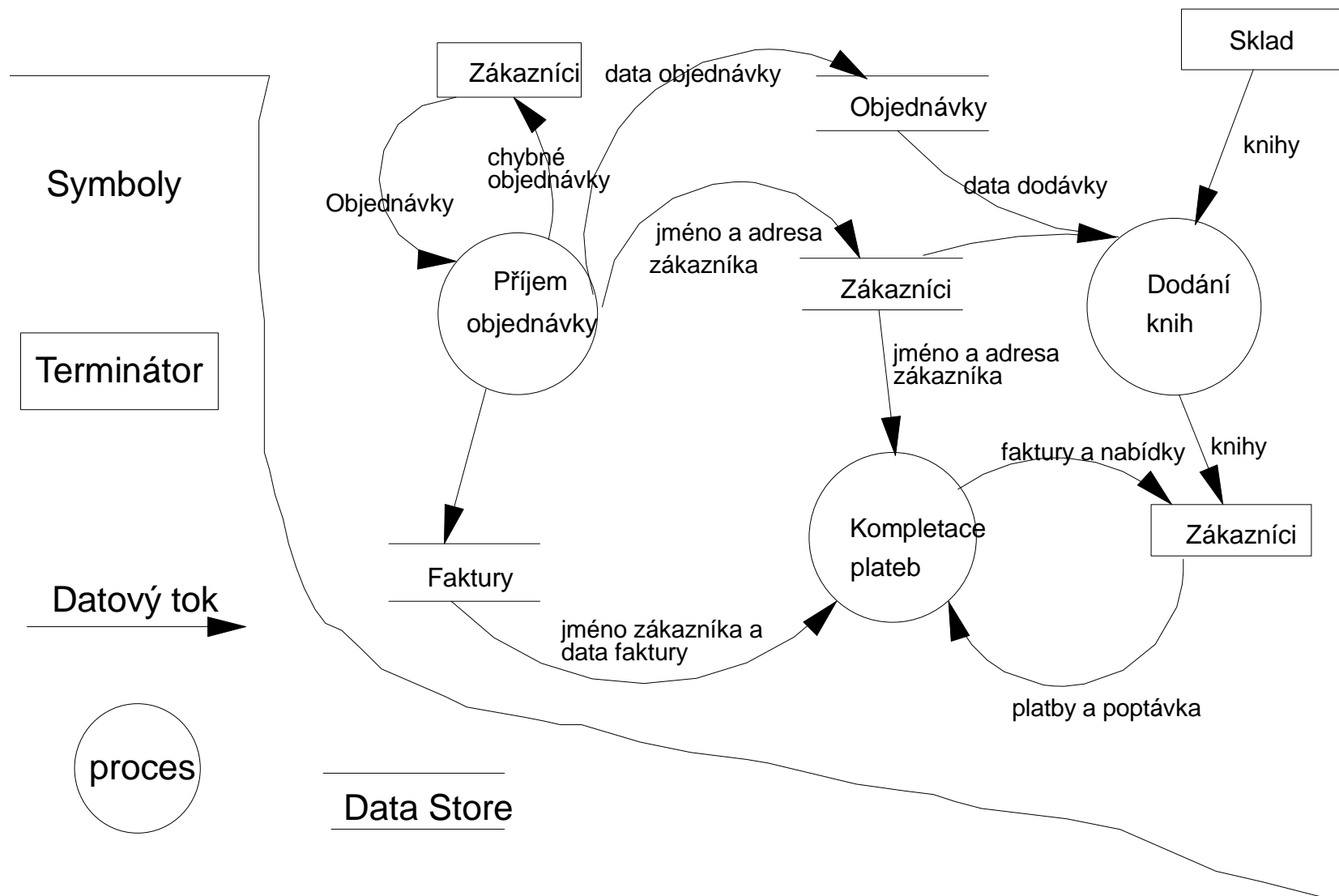
Události a jejich obecný kontext

Model funkcí (Data Flow Diagram)

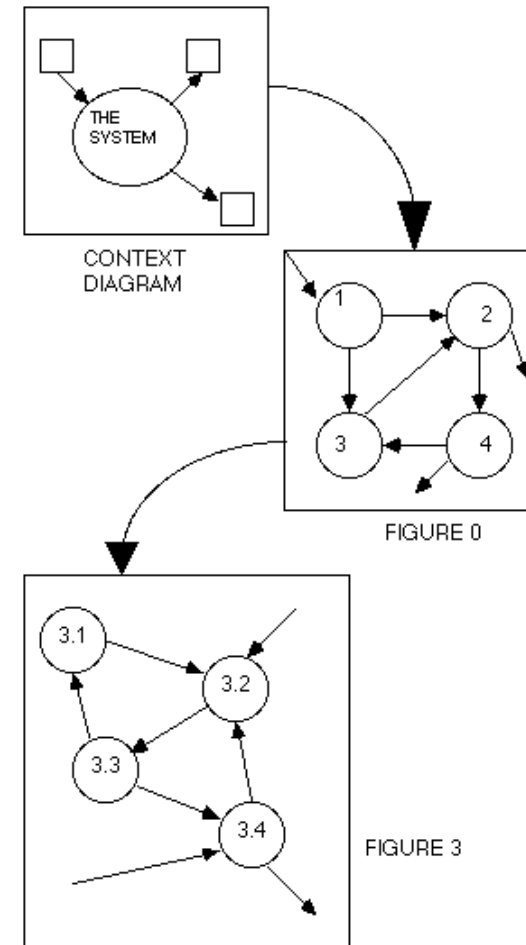
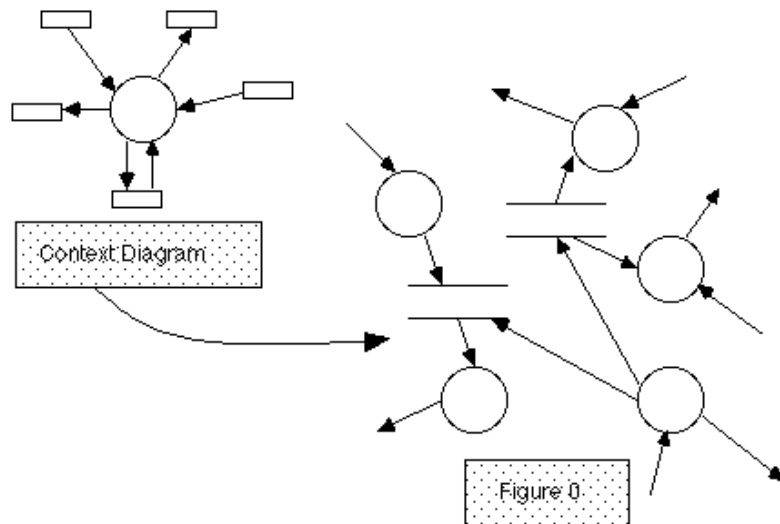
Produkty, vstupy, výstupy, aktéři, business omezení procesů (životní cykly objektů)

Účelové kombinace ŽC objektů, kontext chování objektů

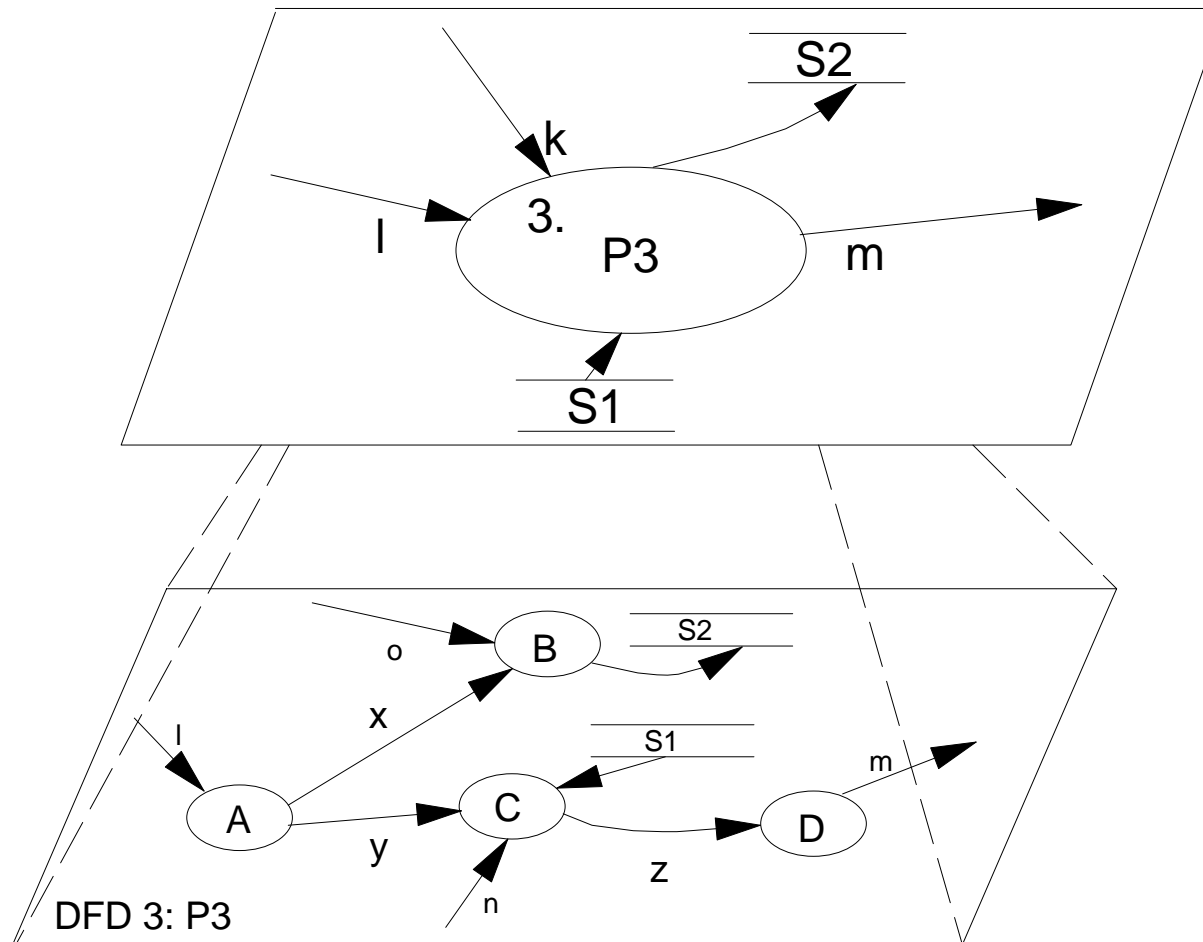
# Data Flow Diagram (DFD)



# Hierarchie DFD



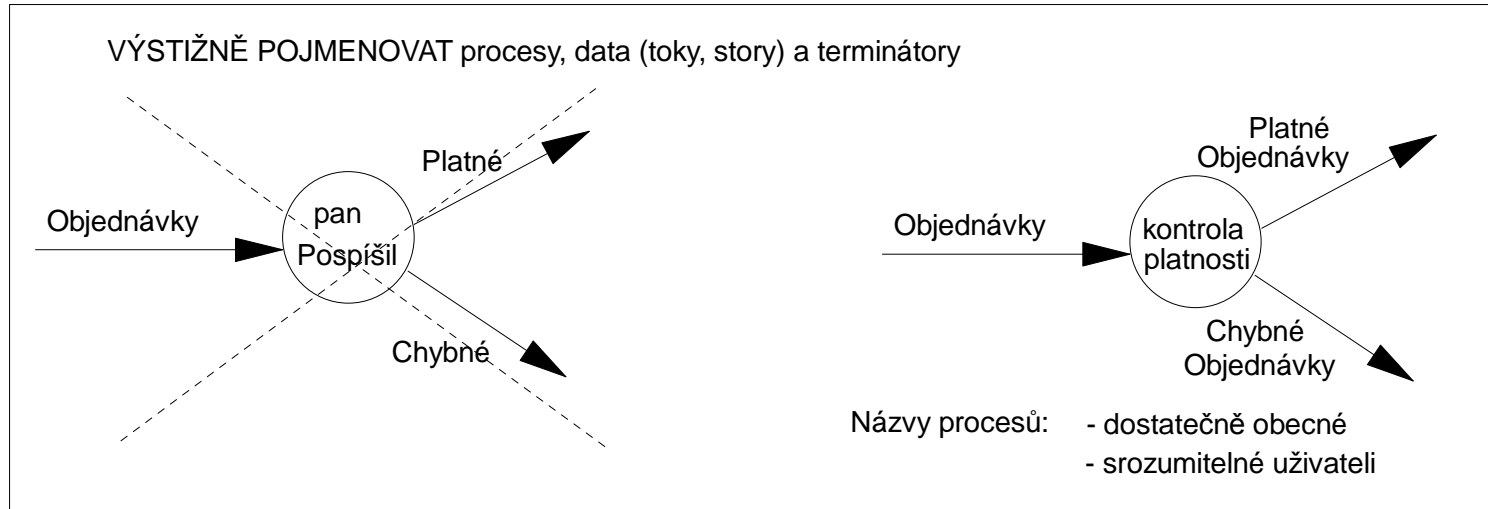
# Konzistence hierarchie DFD



Data Dictionary

⋮  
 $k = n + o$   
 ⋮

# Pravidla tvorby DFD



## OČÍSLOVAT procesy

- číslo identifikuje proces v rámci úrovně
- číslo určuje příslušnost procesu do nadřazeného procesu



## Volit SNESITELNOU SLOŽITOST DFD

- DFD s příliš procesy je nesrozumitelný (rozdělit do úrovní)
- Jeden DFD = 7 +/- 2 procesy

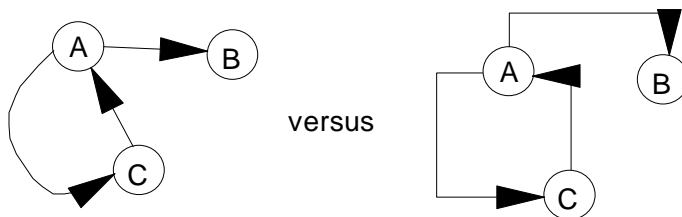
vždy musí být úplný

# Pravidla tvorby DFD



Volit DOSTATEČNĚ ESTETICKÉ USPOŘÁDÁNÍ DFD

- Velikost a tvar bublin
- Oblé versus hranaté spojnice

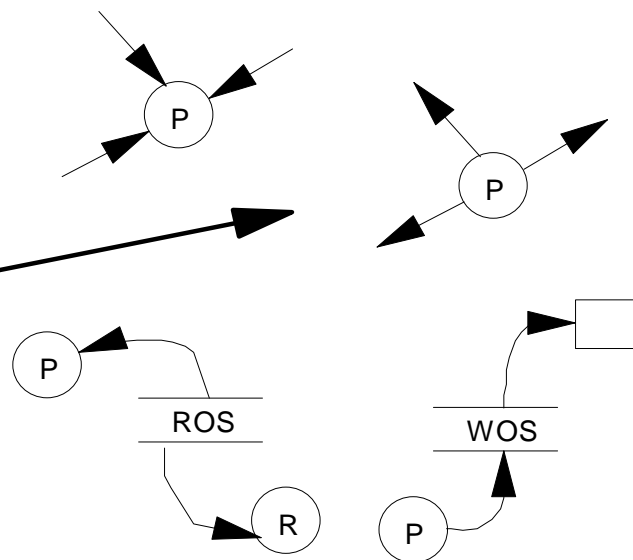


Sledovat FORMÁLNÍ SPRÁVNOST DFD

t.j. soudržnost uvnitř DFD

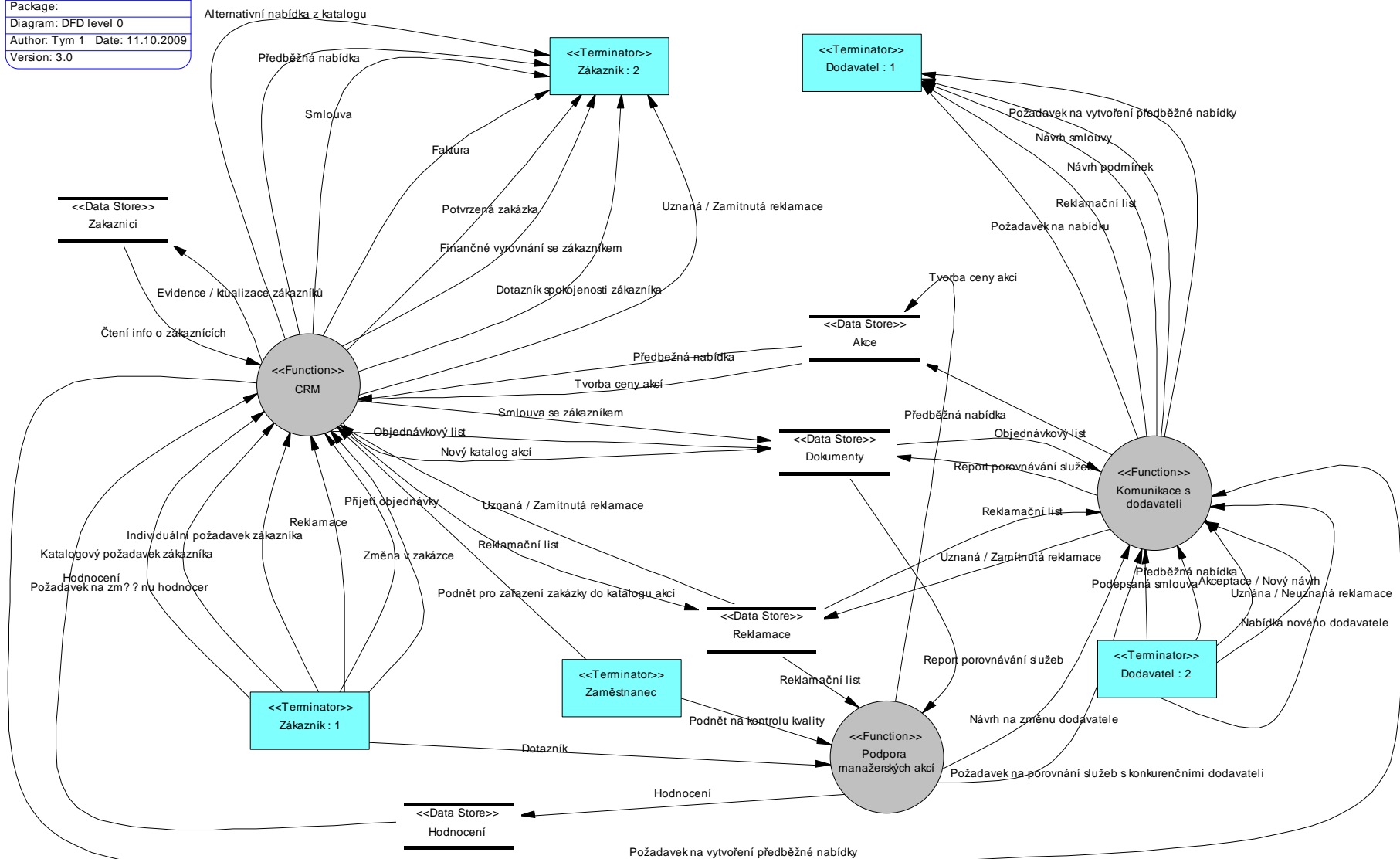
DFD NESMÍ obsahovat:

- ČERNÉ DÍRY
- Samogenerující procesy
- Neoznačené toky a procesy
- READ ONLY a WRITE ONLY story



# Příklad DFD

Object-Oriented Model
Model: Funkční model
Package:
Diagram: DFD level 0
Author: Tým 1 Date: 11.10.2009
Version: 3.0

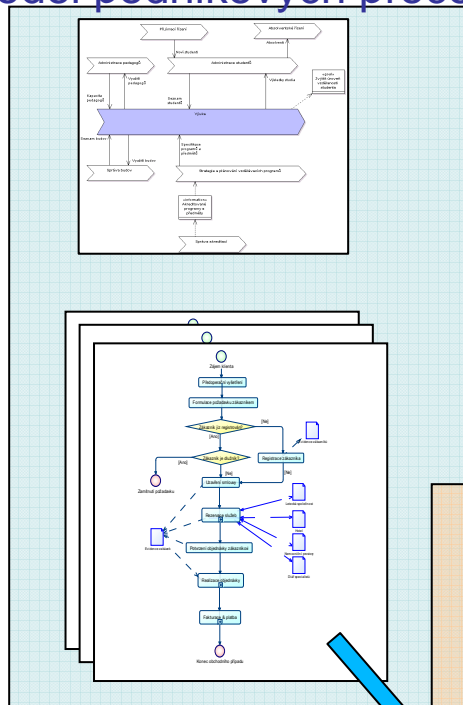




# Jak Data Flow Diagram vytvořit (technika zkoumání událostí)

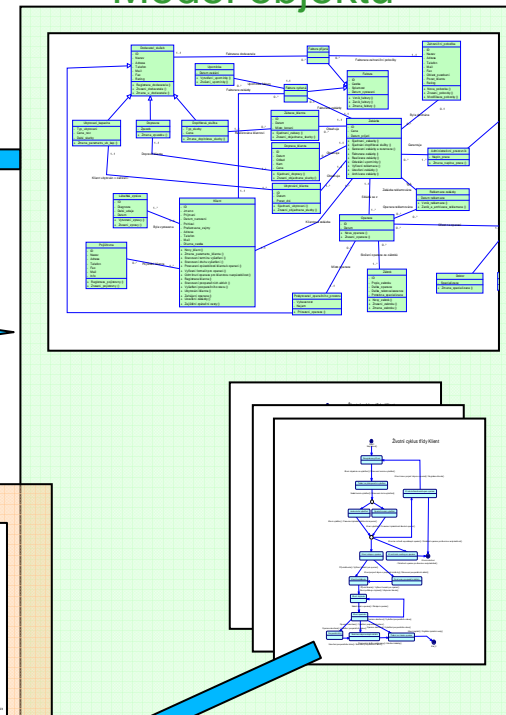
# Data Flow Diagram modeluje relevantní kombinace událostí a akcí

Model podnikových procesů



Události a jejich účelové kombinace

Model objektů

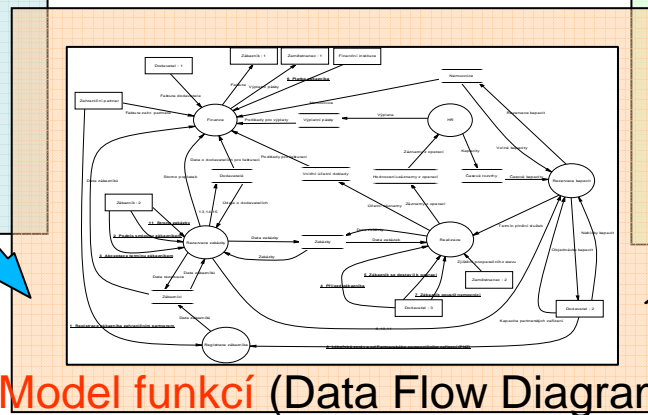


Události a jejich obecný kontext

Model funkcí (Data Flow Diagram)

Produkty, vstupy, výstupy, aktéři, business omezení procesů (životní cykly objektů)

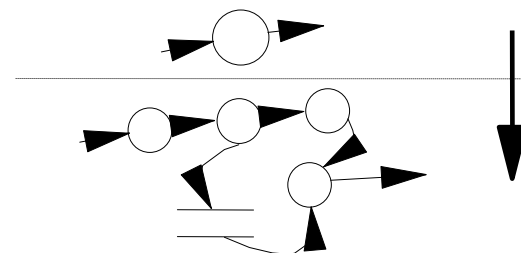
Účelové kombinace ŽC objektů, kontext chování objektů



# MOŽNOSTI TVORBY FUNKČNÍHO MODELU

■ *Top-down funkční dekompozice*

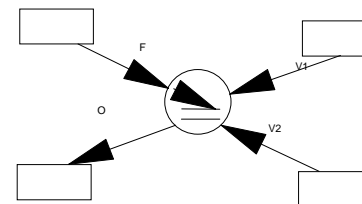
nebo



■ *Dle výstupních datových toků (výstupy -> procesy -> vstupy)  
+ kompozice vyšších úrovní a dekompozice na nižší úrovně*

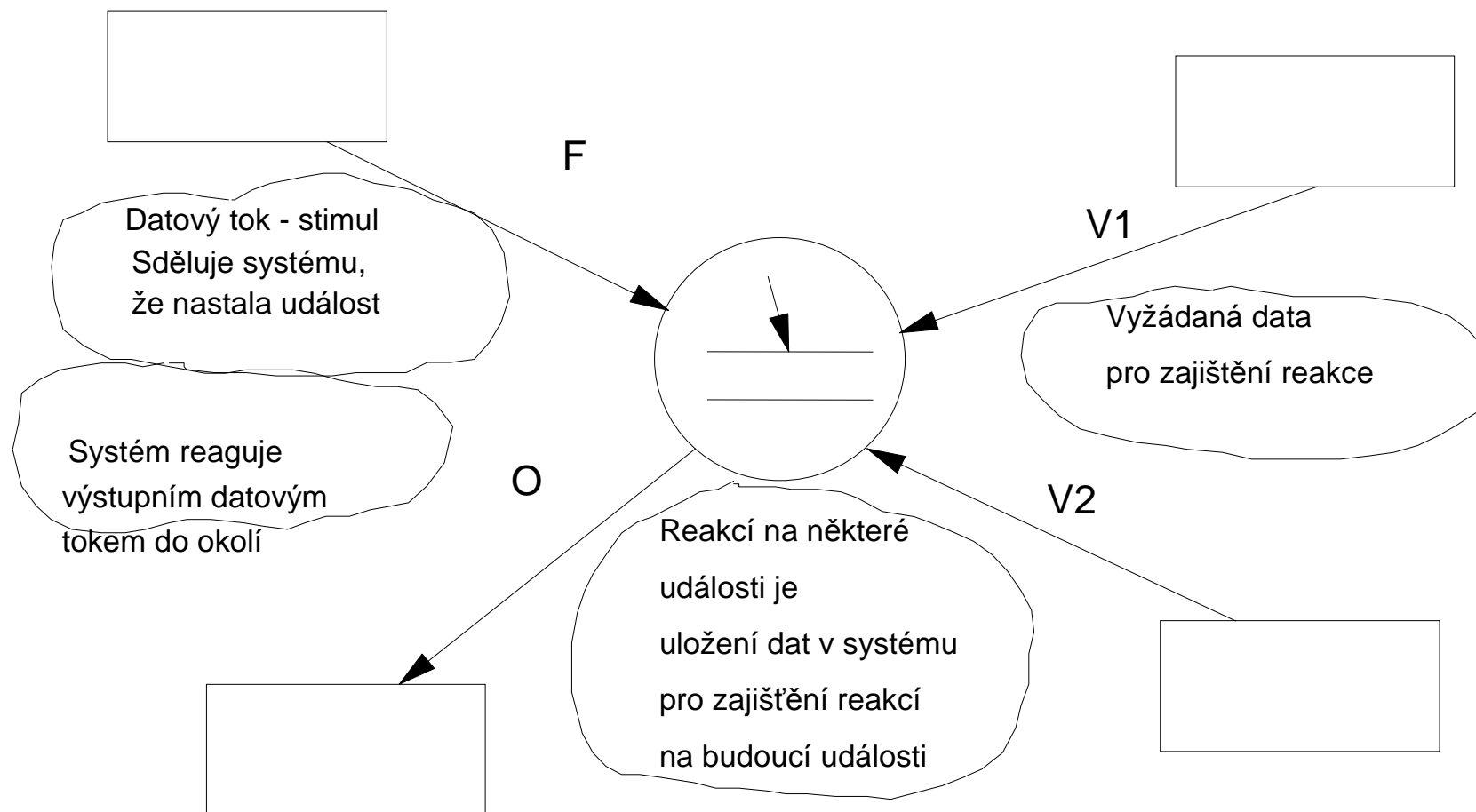
nebo

■ *Dle událostí (událost -> proces -> vstupy+výstupy)  
+ kompozice vyšších úrovní a dekompozice na nižší úrovně*



zdroj: YSM

# Událost -> stimul -> reakce



zroj: YSM

(c) USKA, 1994

# Event Partitioning Approach



Pro každou UDÁLOST vytvořit PROCES



Každý PROCES pojmenovat podle REAKCE systému na událost



Ke každému procesu doplnit VSTUPY a VÝSTUPY a případně DATA STORY.

"Jaká data funkce potřebuje, co je jejím výstupem ?"



**KONTROLA KONSISTENCE**

t.j. balancování výsledku s kontextovým diagramem.

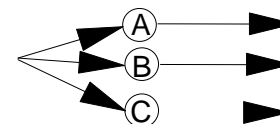


**KOMPOZICE MEZIÚROVNĚ** (úroveň 0)  
přístupem INFORMATION HIDING (skrývání Data Storů)

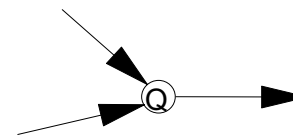
- mezi funkcemi vyhledat LOKÁLNÍ DATA STORY
- tento DS se svými funkcemi tvoří FUNKCI VYŠŠÍ UROVNĚ (vytvoření diagramu vyšší úrovně)
- rozpustit původní diagram do SUBDIAGRAMŮ

Poznámky:

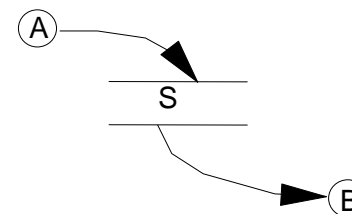
- \* 1 událost & různé reakce



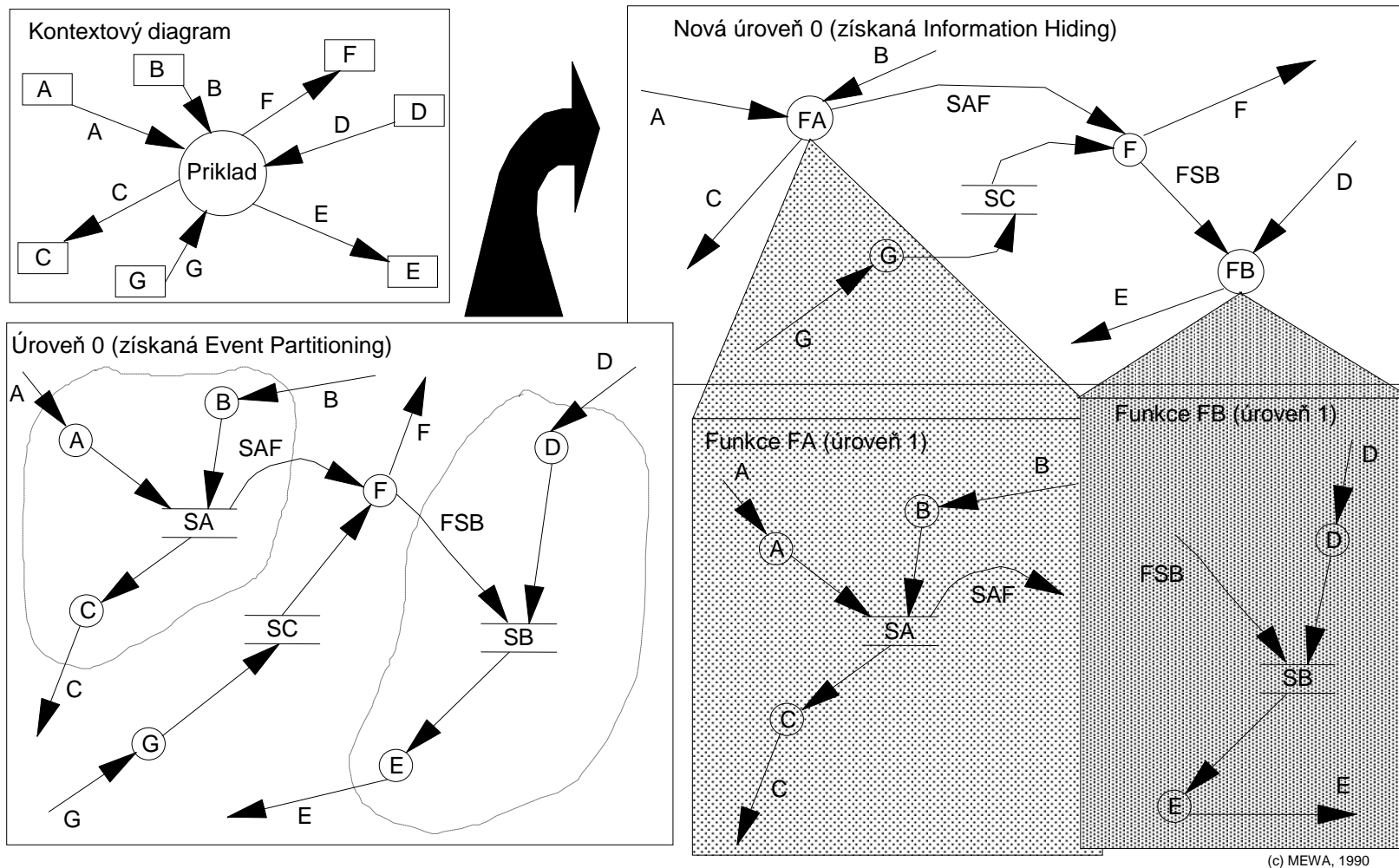
- \* Více událostí & shodná reakce



- \* Data Store je PŘIROZENOU FORMOU komunikace nesynchronizovatelných procesů. Jde o ESENCIÁLNÍ DS !



# Postup tvorby funkčního modelu ( Kompozice meziúrovně - Information Hiding )



(c) MEWA, 1990

(c) MEWA, 1994

Zač Data Flow Diagram považovat  
(„fyzikální“ podstata DFD)

# Realizace DFD v prostředí UML

Specializace diagramu tříd, 4 standardní stereotypy:

prvek DFD	metatřída UML	specializace (základní omezení) metatřídy
DataStore	class	standardní metody čtení, zápisu a zrušení (destructor)
Funkce	class	metody irelevantní
Terminátor	class	metody irelevantní
DataFlow	association	orientovaná, jedno, či obousměrná

Pravidla konsistence DFD (konsistenční omezení vztahů metatříd):

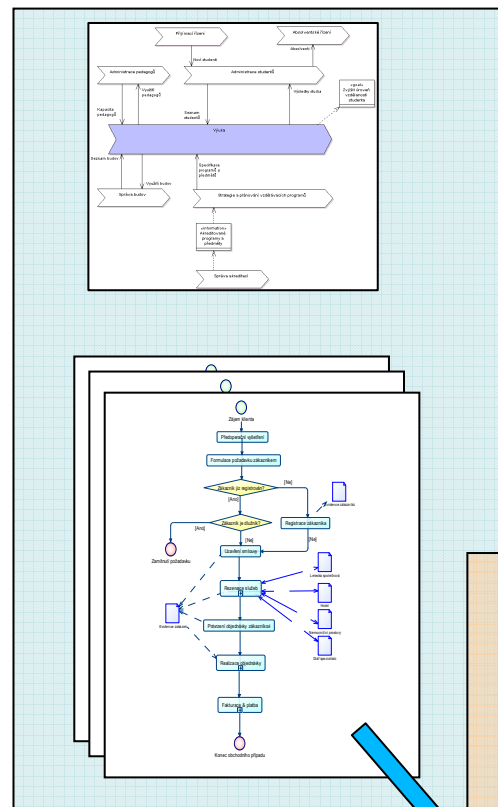
- DataStore musí mít alespoň jeden vstupní DataFlow a jeden výstupní DataFlow.
- DataFlow smí spojovat pouze Funkci a Funkci, Funkci a DataStore nebo Terminátor a Funkci.
- DataFlow Terminátor -> Funkce musí mít přiřazenu událost
- Funkce musí mít alespoň jeden DataFlow



# Konsistence DFD s ostatními analytickými modely

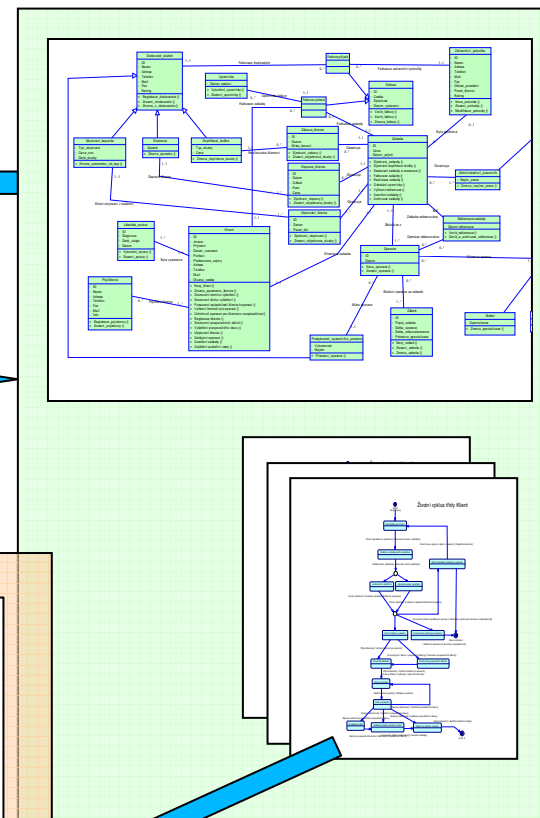
# Přehled analytických modelů

**Model podnikových procesů**  
(Globální model procesů, Process Diagrams)



Události a jejich  
účelové  
kombinace

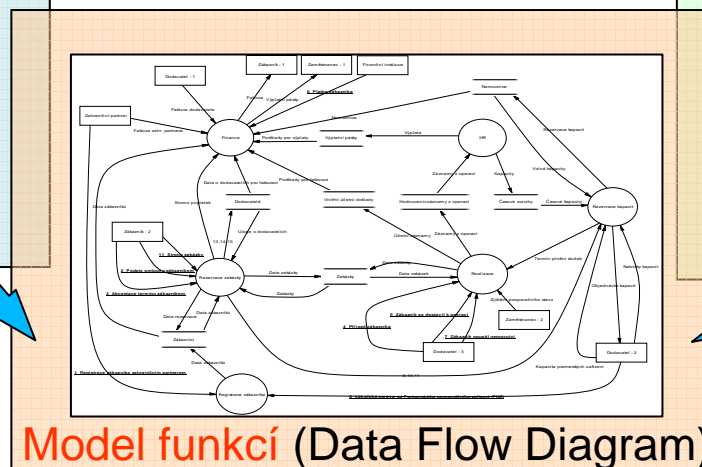
**Model objektů**  
(Class Diagram, State Charts)



Události a jejich  
obecný  
kontext

Produkty, vstupy, výstupy,  
aktéři, business omezení  
procesů (životní cyklus  
objektů)

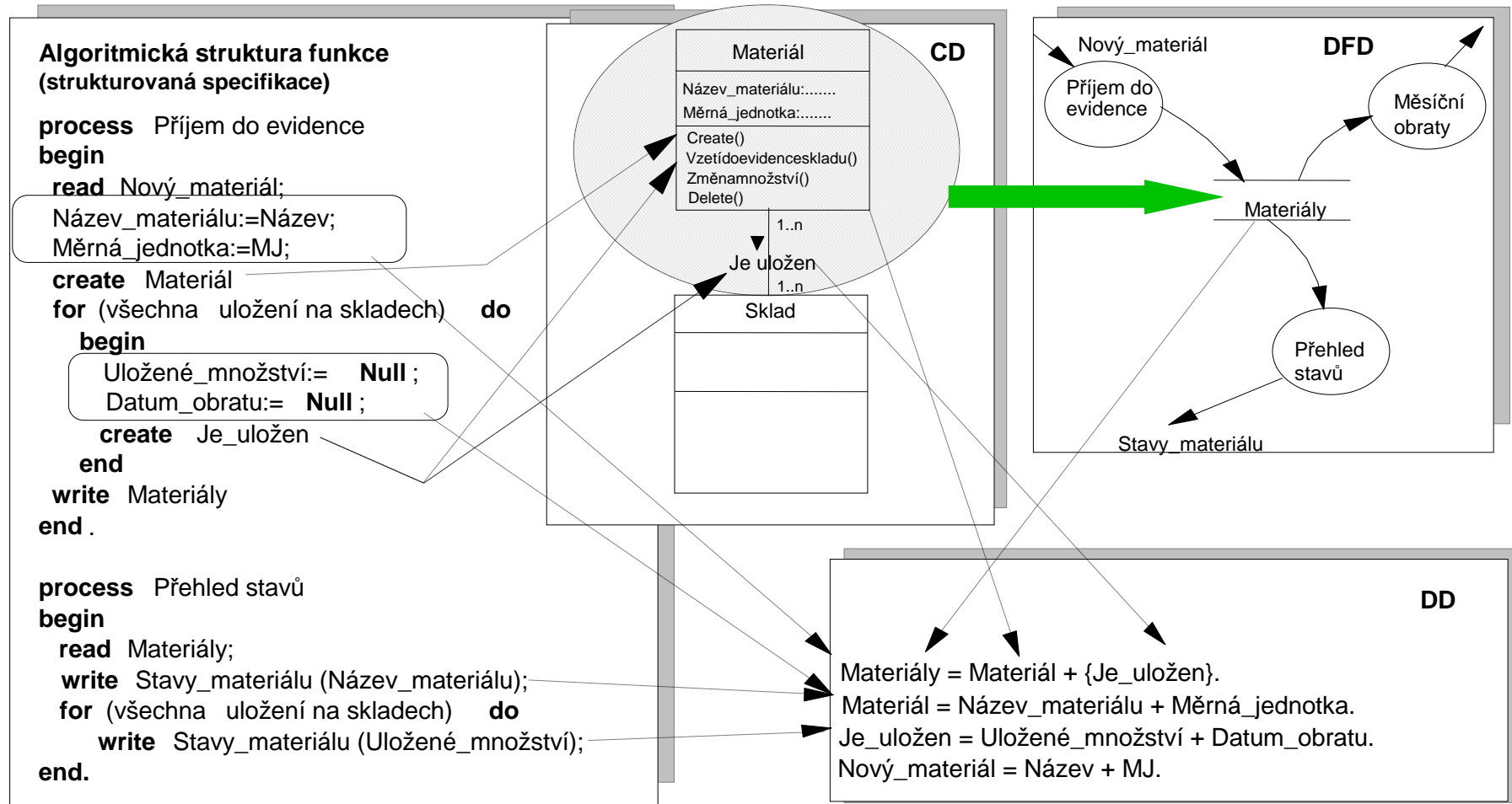
Účelové kombinace ŽC objektů,  
kontext chování objektů



# Provázání DFD s objekty

- Každý elementární Datastore v DFD musí být v CD zastoupen jako třída, nebo asociace, anebo kombinace obojího.
- Atributy každého elementárního Datastore z DFD musí být datovou strukturou atributů tříd, jimiž je tento Datastore v CD zastoupen.
- Metody každé elementární funkce z DFD musí být algoritmickou strukturou metod tříd, jimiž jsou v CD zastoupeny Datastory, spojené datovými toky s touto funkcí

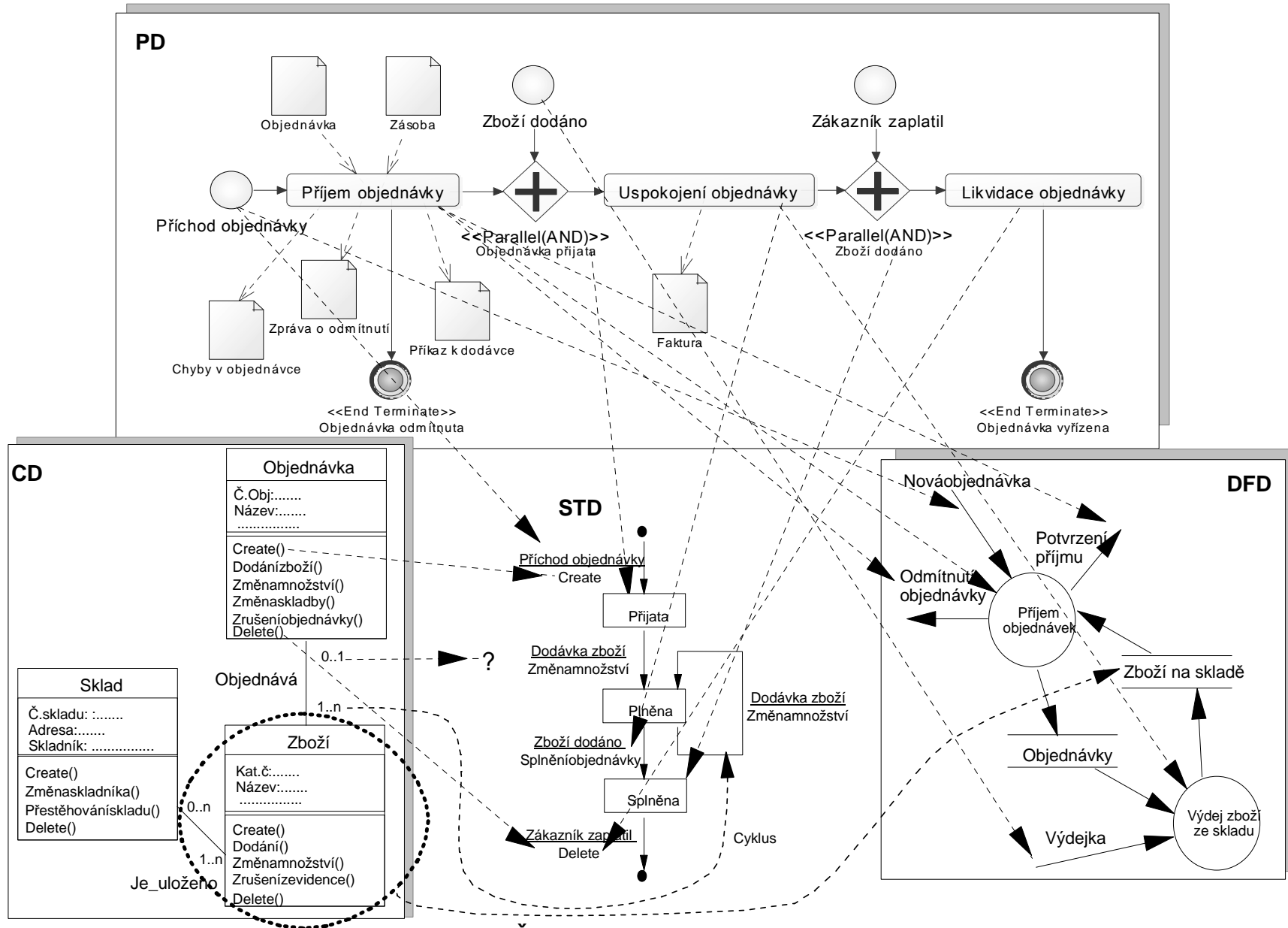
# Příklad provázání DFD s objekty



# Provázání DFD s procesy

- Každý proces má vazbu alespoň na 1 funkci
- Každá funkce má vazbu alespoň na 1 proces
- Každá událost v procesním modelu má vazbu na vstupní tok v DFD
- Každý elementární vstupní datový tok v DFD od terminátoru (tj. zvnějšku systému) musí odpovídat nějaké události, specifikované v popisu nějakého (nějakých) business procesu (procesů) v PD.
- Každý stav každého procesu v PD musí korespondovat s některým(i) elementárním(i) Datastorem(y) v DFD a naopak každý elementární Datastore v DFD musí korespondovat s některým(i) stav(y) procesů(ů) v PD. Jde o korespondenci M:N.

# Ilustrace vztahů mezi procesním, objektovým a funkčním modelem



# Závěr

- Diagram datových toků je svou „fyzikální“ podstatou konceptuálním modelem funkcí informačního systému.
- Jedná se o model strukturální (tedy objektové povahy, nikoliv povahy procesní, jak se vždy myslelo)
- Cílem je popsat funkčnost informačního systému
  - funkčnost = potenciál chování systému
  - cílem chování informačního systému je odrážet dění v reálném (business) systému – viz Princip modelování
  - neboli modelovat relevantní kombinace událostí a akcí
    - ať obecně platné (neb nutno respektovat obecná „business rules“)
    - tak záměrné (neb nutno podporovat „business procesy“)
- Paralela k metodickému vývoji v oblasti objektově orientované analýzy - obecné klasifikace typů analytických objektů:
  - funkční objekty (alias funkce),
  - datové objekty (alias DataStory)
  - business objekty (terminátory / konceptuální entity)