

# Funkční analýza

Předmět Informační systémy

Daniela Szturcová

# Projektování IS

- IS má za účel zefektivnit práci s informacemi.
- Při projektování IS zohledňujeme potřeby zákazníka, definujeme firemní procesy a teprve poté přistupujeme k návrhu IS.
- Používané postupy:
  - analýza
  - syntéza

# Analýza systémů

[systems] analysis is frustrating, full of complex interpersonal relationships, indefinite, and difficult. In a word, it is fascinating. Once you're hooked, the old easy pleasures of system building are never again enough to satisfy you.

Structured Analysis and Systems Specification

(DeMarco, 1978)

# Proč analyzujeme?

- Porozumění řešeného problému.
- Snaha o standardizaci pracovních postupů.
- Zlepšení komunikace - v týmu řešitelů, mezi řešiteli a zákazníkem či uživatelem.
- Zachycení analytických postupů ve formě dokumentace - aktuálnost, úplnost, možnost opakovat.

# Analytické přístupy

Objektově – orientovaná analýza a návrh

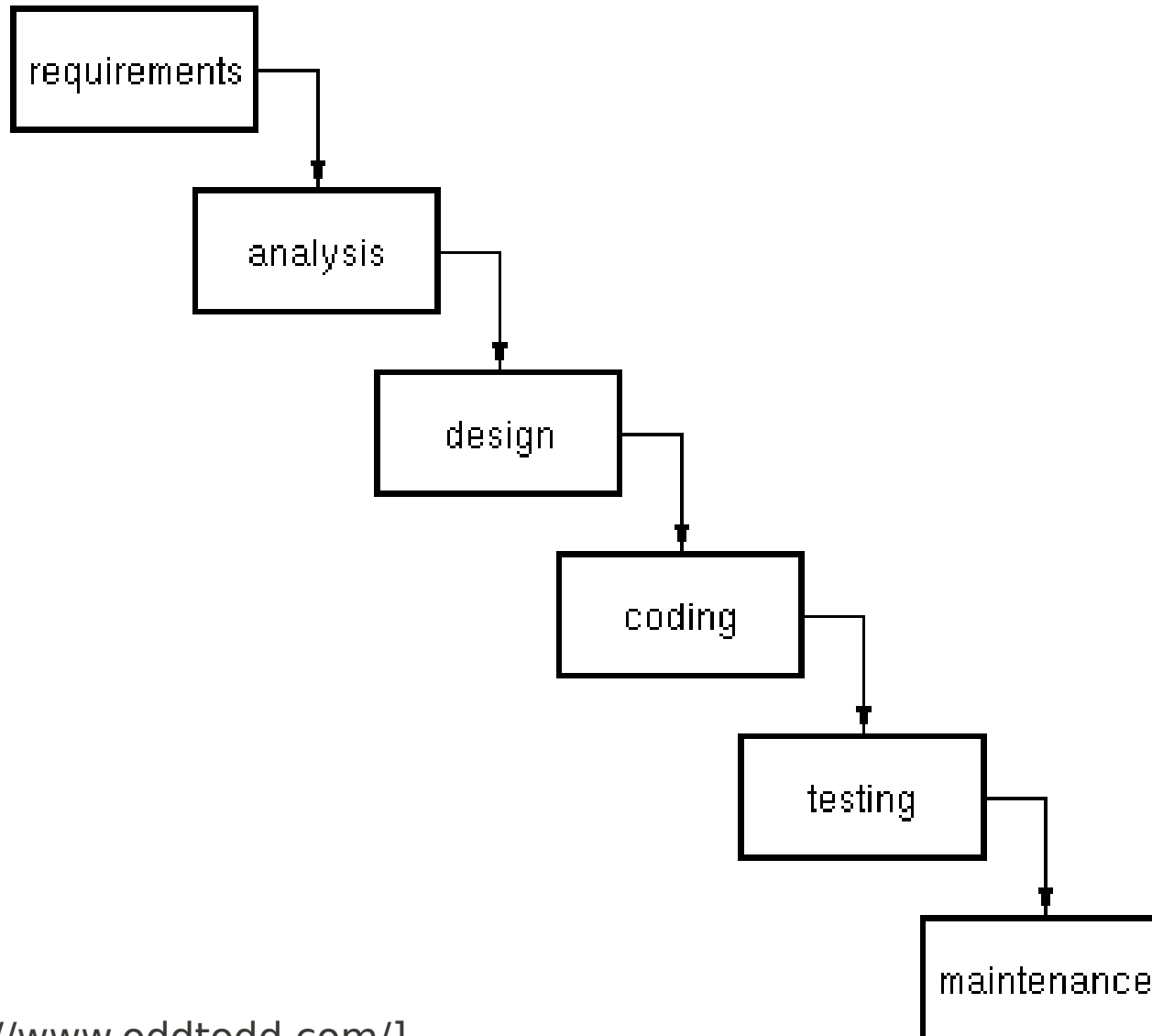
Strukturovaná analýza a návrh

- Odlišný filosofický princip
- Vývoj a tvorba IS je podmíněna výběrem přístupu a použitou metodikou

# Strukturovaná analýza a návrh

- Odděluje data a procesy
- Vznikají datové a funkční modely, které spolu korespondují
- Většinou se používá vodopádový model pro vývoj IS a s tím spojený postup dekompozice systému

# Vodopádový model



[<http://www.oddtodd.com/>]

# Specifikace požadavků

- První fáze – snaha tvůrce systému zmapovat potřeby uživatelů systému.
- Představuje úplný popis chování systému, který budeme vyvíjet.
- Tvoří se jako sada slovních popisů či diagramů, které zachycují všechny interakce uživatele s vytvářeným systémem.



# Specifikace požadavků

- Interview
- Legislativa - doména v obecné rovině, tak u zákazníka
- Opakování ověřených postupů založených na zkušenosti tvůrce
- Výstupem by měl být oboustranně odsouhlasený a podepsaný dokument

# Funkční analýza

- Zaměřuje se na funkce systému – identifikuje je a popisuje.
- Definuje a popisuje také
  - události a
  - transakce.
- Výsledkem je funkční model systému.
- Koresponduje s dalšími modely – datovým, časovým.

# Seznam událostí

ID	Název událostí	Typ	Reakce systému
1	Stávník nabíjí svůj účet	Datový	Přijme peníze, vystaví nový stav účtu
2	Strávník si prohlíží stav, historii účtu	Datový	Zobrazí stav účtu
3	Strávník si prohlíží jídelníček	Datový	Zobrazí aktuální jídelníček
4	Strávník si objednává jídlo	Datový	Vystaví potvrzení o objednávce, přepočte nabídku jídel
5	Strávník ruší objednávku jídla	Datový	Provede přepočet jídel v nabídce
6	Strávník si prohlíží seznam objednávek	Datový	Zobrazí stav objednávek
7	Je poslední den v měsíci	Časový	Provede se odečet z platu zaměstnance
8	Správce zakládá/ruší účty strávníků	Datový	Založí/zruší účet, obnoví stav účtů
9	Správce vede evidenci zaměstnanců	Datový	Založí/změní/zruší účet, obnoví stav zaměstnanců

# Funkční diagramy

V literatuře se dají nalézt pod různým označením:

- procesní model (process model),
- diagram toku práce (work flow diagram),
- funkční model (function model),
- bublinový diagram (bubble chart or diagram),
- diagram datových toků, toků dat (DFD).

# Diagram datových toků - DFD

## DFD - Data Flow Diagram

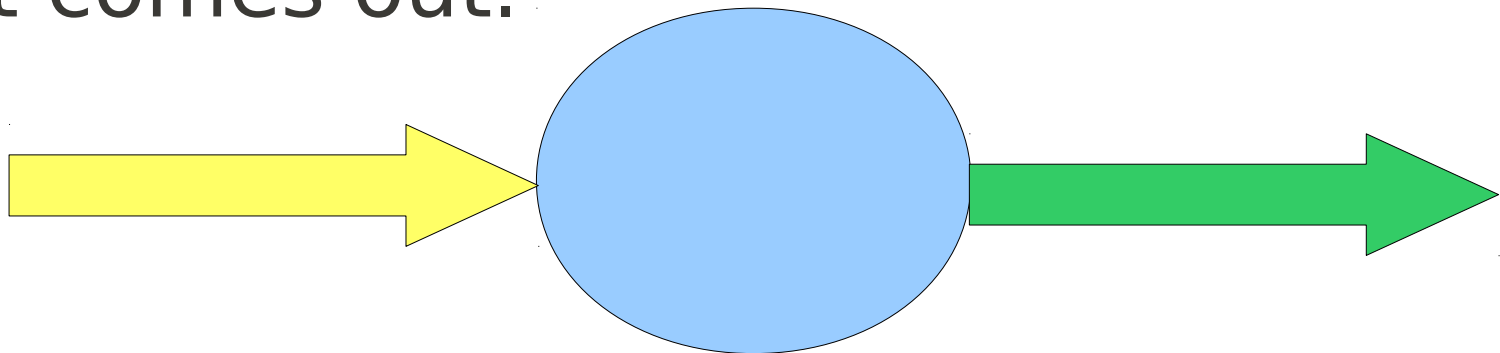
- Z historie jsou známy první pokusy znázornění datových toků v organizační struktuře podniku a výroby již na počátku 20. století.
- Dnes patří diagramy datových toků mezi nejužívanější nástroje strukturované analýzy.

# Define DFD

A data flow diagram (DFD) is a graphical representation of the flow of data between processes.

In other words, it shows:

- What goes in.
- How it is changed.
- What comes out.



# Definice dle De Marca

- Diagram datových toků je síťovou reprezentací systému.
- Systém může být automatizovaný, manuální nebo smíšený.
- DFD znázorňuje systém pomocí jeho komponent a určuje rozhraní mezi komponentami.

# DFD

DFD se skládá ze čtyř základních prvků, kdy se každá komponenta znázorňuje pomocí grafického symbolu.

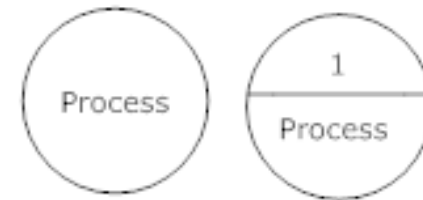
- Proces
  - Datová paměť
  - Datový tok
  - Aktér (terminátor)
- Každý grafický symbol je označen názvem, který charakterizuje danou komponentu.



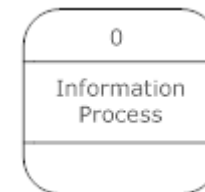
# Process

A process transforms incoming data flow into outgoing data flow.

Yourdon and Coad Process Notations



Gane and Sarson Process Notation



# Proces

- Proces představuje část systému, která provádí transformaci vstupů na výstupy. Název má vystihovat podstatu funkce.
- Vstupní a výstupní data mají odlišná pojmenování i v případě, že se jedná o data se stejnou strukturou.



# DataStore

Datastores are repositories of data in the system. They are sometimes also referred to as files.

Yourdon and Coad Process Notations



Gane and Sarson Process Notation



# Datová paměť (datový sklad)

- Data Store znázorňuje uchovávaná data (materiál).
- Paměť může realizována na počítači i jinou formou (kartotéka lístků, objednané zboží na skladě, ...).
- Paměti představují místa, kde jsou po nějakou dobu uložena data, bez porušení jejich struktury, a odkud o nich mohou být čerpány informace.

---

Produkty

---

# Datová paměť

Důvody existence datových pamětí:

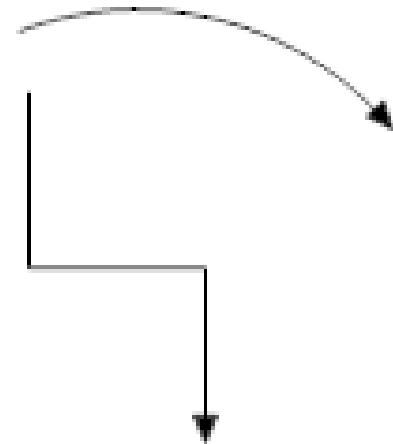
- požadavek uživatele (data předávaná mezi procesy, které pracují v různých časových obdobích),
- důsledek implementace (mezivýsledky při postupné transformaci dat).

Pravidlo: do každé paměti alespoň jeden datový tok vstupuje a jeden vystupuje.

# Dataflow

Dataflows are pipelines through which packets of information flow. Label the arrows with the name of the data that moves through it.

Dataflows Notations



# Datový tok

- Datový tok (Data Flow) popisuje přesun skupiny údajů mezi jednotlivými částmi systému nebo mezi systémem a okolím.
- Na rozdíl od paměti reprezentuje data v pohybu.
- V případě obecného modelu může tok dat představovat i pohyb hmoty (materiál, zboží, ...).
- Názvy jsou voleny v souladu s typem přenášených informací.



Detaily o produktu

# External Entity

- External entities are objects outside the system, with which the system communicates. External entities are sources and destinations of the system's inputs and outputs.

External Entity Notations





# Aktér (Terminátor )

Aktér představuje zdroj a/nebo příjemce dat v okolí v okolí systému. Mohou to být osoby, objekty, případně i jiné systémy mimo analyzovanou oblast.

Lze je identifikovat dle rolí, které jsou vymezeny různými uživateli systému.

V rámci analyzované oblasti to jsou objekty nezávislé na modelovaném systému, ale se systémem komunikují.

Obchodní oddělení

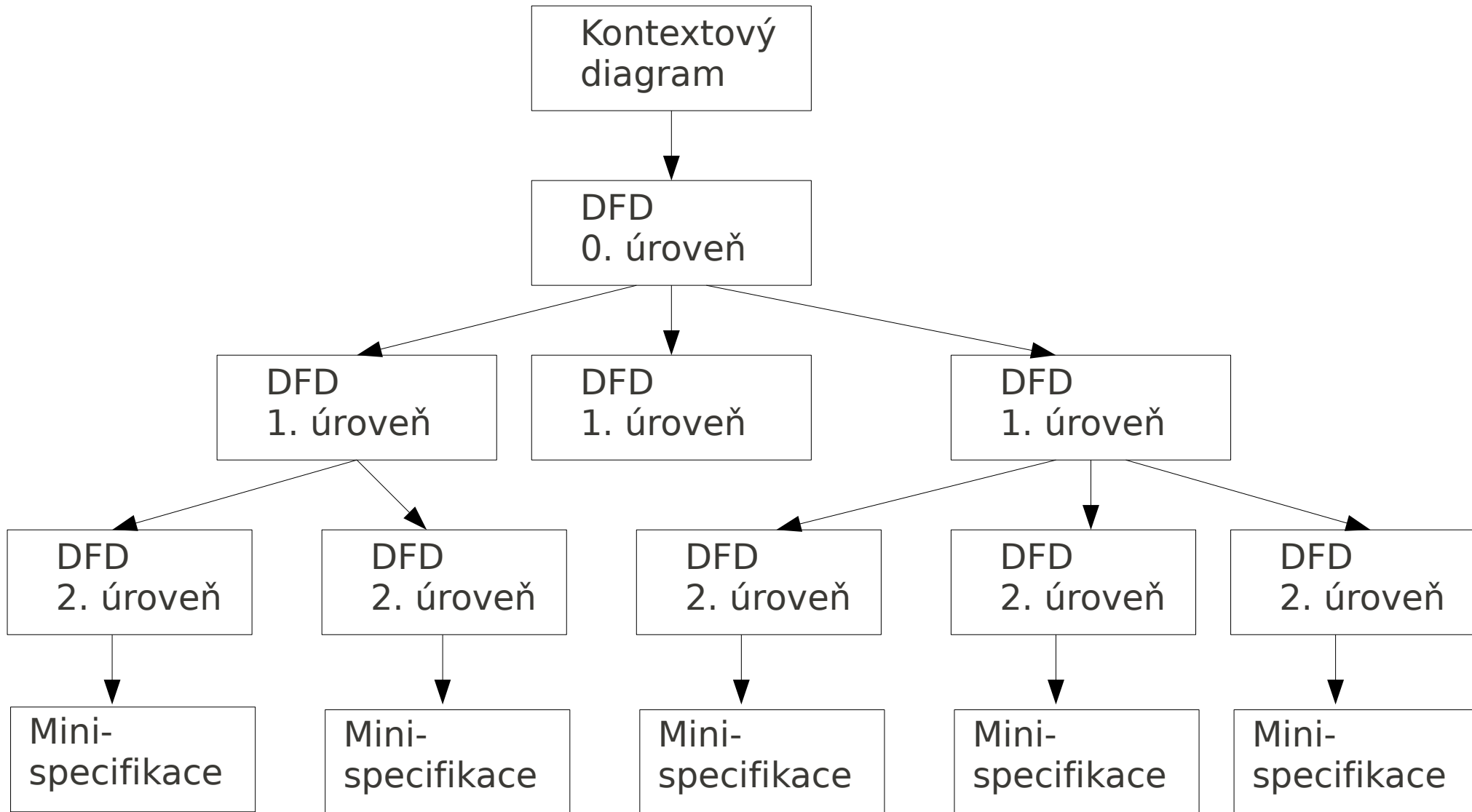
# Pravidla pro aktéry

- Aktér se nachází mimo modelovaný systém.
- Tok mezi aktérem a procesem charakterizuje rozhraní systému.
- Chování aktérů je nezávislé na analytikovi, nezkoumáme je.
- Pokud se vztahy mezi aktéry ukáží důležité pro chod systému – zahrneme je do modelu jako procesy, změníme strukturu DFD.

# Postup Top-Down

- Rozklad shora dolů
- Nejvyšší úroveň – kontextový diagram.
- N-tá úroveň – rozklad na každé úrovni zpodobňuje procesy z vyšší úrovně.
- Vytváří se hierarchická reprezentace.
- Systém reprezentujeme pomocí stromové struktury, kdy kořenem stromu je nejvyšší úroveň modelovaného systému.

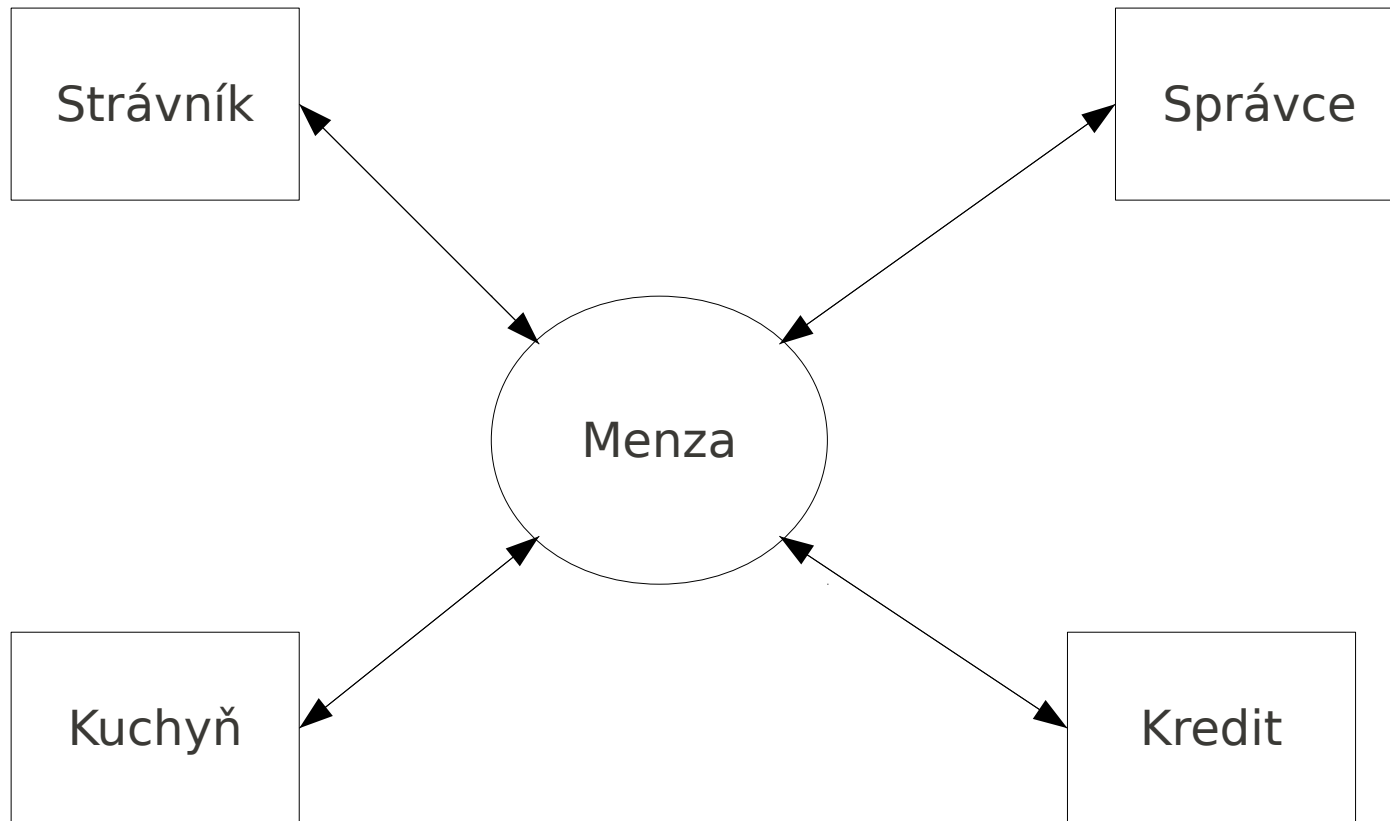
# Hierarchie diagramů



# Kontextový diagram

- Má zvláštní postavení mezi DFD, často je zařazován mezi modely vnějšího chování systému.
- Je zobrazen jako jediný proces s návazností na vnější svět, zobrazují se zde hranice systému a všechny typy terminátorů.

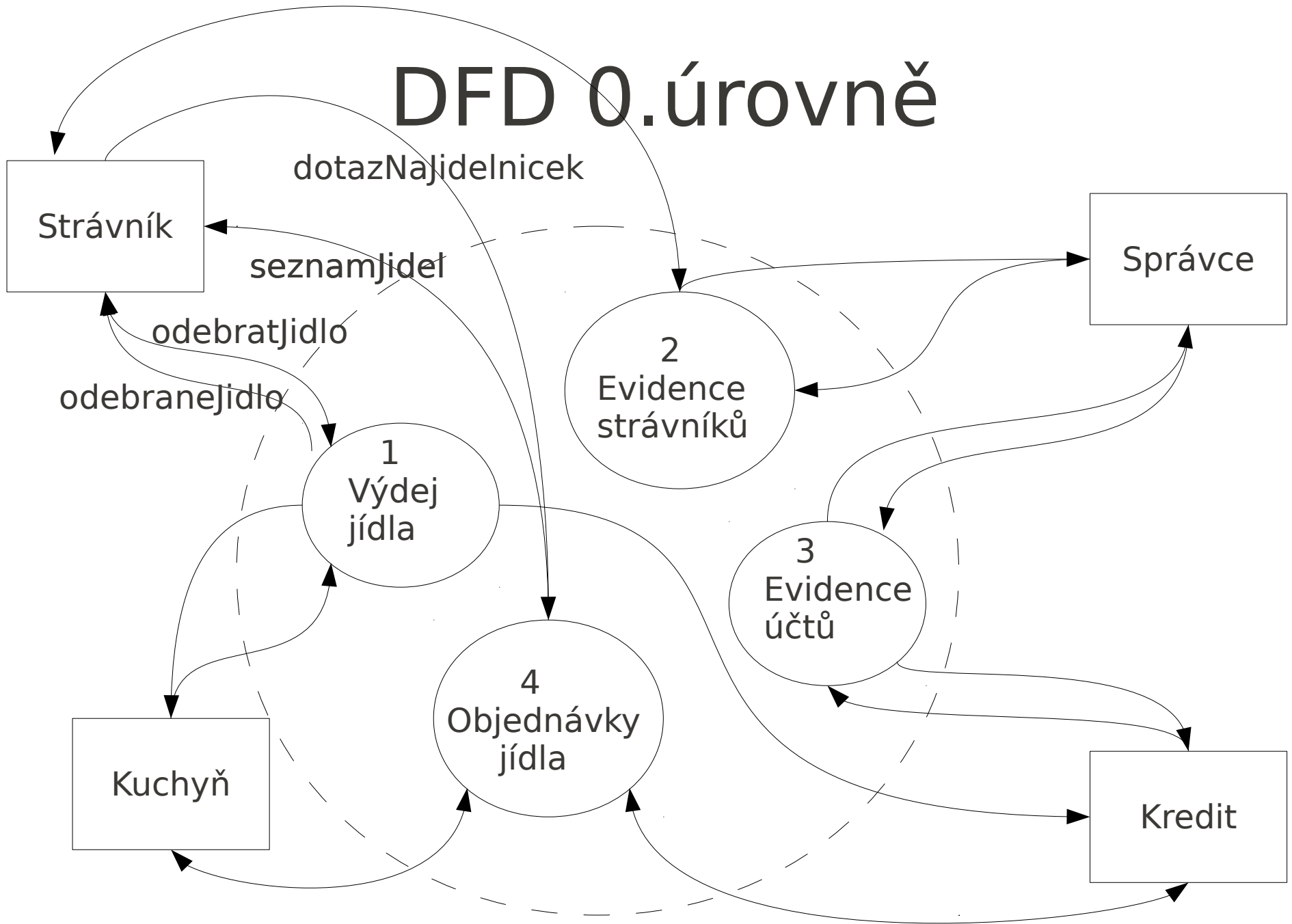
# Kontextový diagram



# DFD úrovně 0

- DFD systému na 0. úrovni rozloží systém na několik hlavních subsystémů.
- V celé reprezentaci rozkladu se doporučuje dodržovat takzvané pravidlo sedmi (jeden diagram zobrazuje nejvýše 7 hlavních subsystémů).

# DFD 0.úrovně

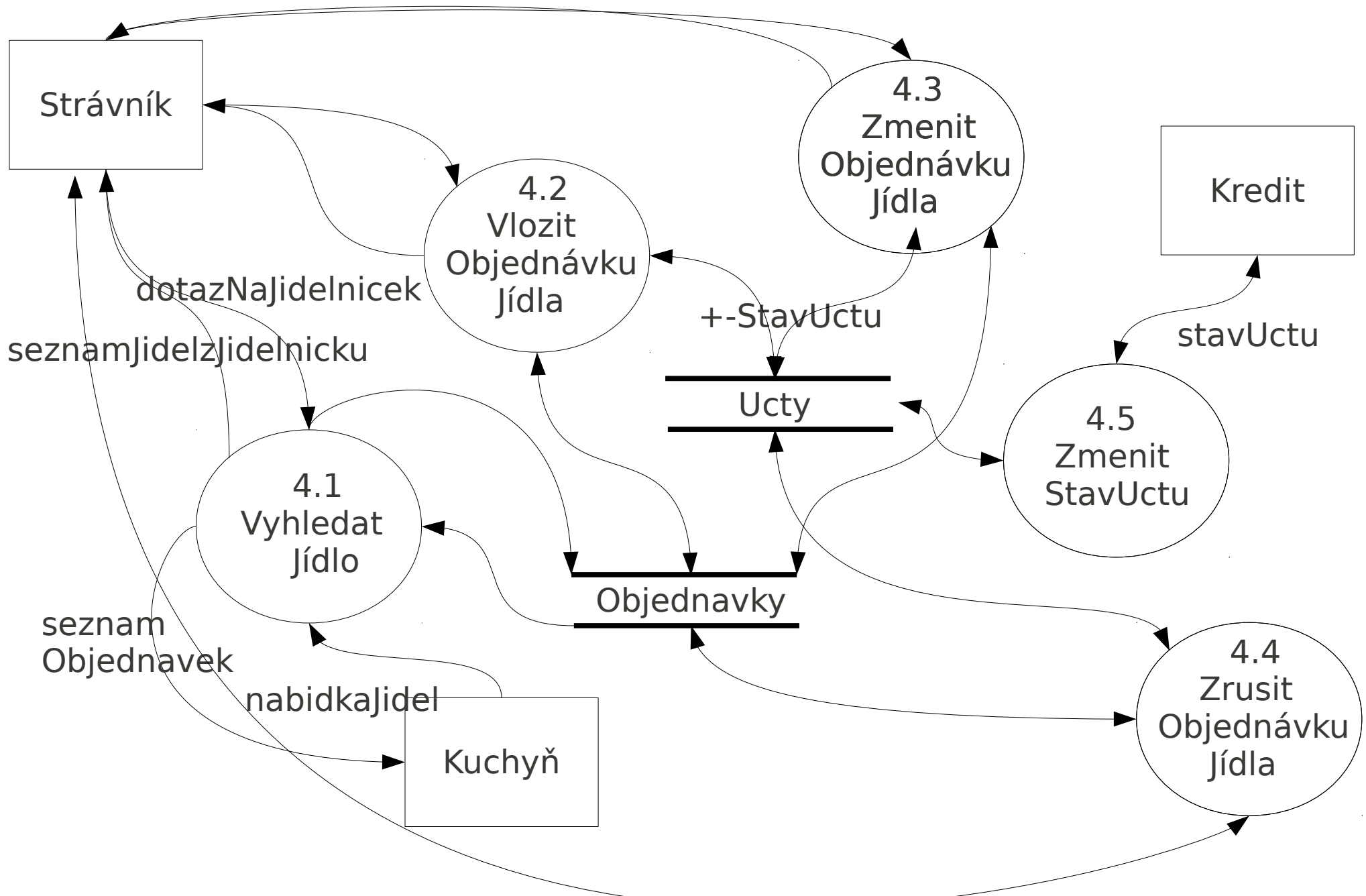




# DFD úrovně 1

- DFD na úrovni 1 znázorňují podrobnější popis každého z hlavních subsystémů z úrovně 0 (jejich počet je tedy dán počtem rozeznávaných procesů z předešlé úrovně).

# DFD 1.úrovně Objednávky jídla



# DFD úrovně 2

- DFD úroveň 2, které podrobněji popisují jednotlivé funkce subsystémů z úrovně 1 (při dodržení pravidla 7 by jich nemělo být více než 49).
- Rozklad na nižší úrovně pokračuje obdobným způsobem až do té doby, kdy popisované procesy dosáhnou takové jednoduchosti, že je možné je popsat pomocí takzvaných minispecifikací.

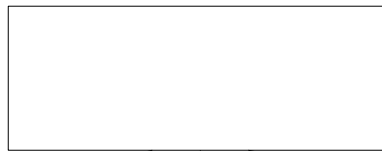
# Pravidla pro tvorbu DFD

Číslování procesů:

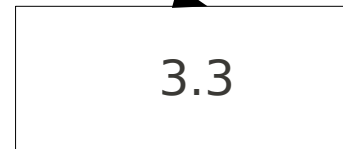
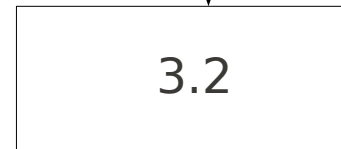
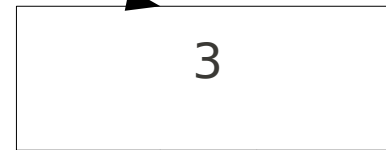
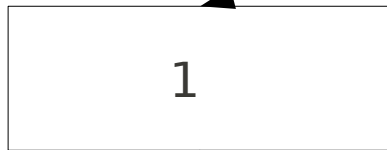
- kontextový diagram bez čísel,
- procesy na úrovni 0 se libovolně očíslovají, a toto označení se dodržuje u všech dalších úrovních,
- počet číslic určuje úroveň rozkladu,
- dodržování číslování jednoznačně určuje větev, do které daný proces patří.

# Číslování DFD diagramů

DFD 0. úroveň



DFD 1. úroveň



# Pravidla pro tvorbu DFD

- Názvy procesů - stručné a výstižné, jednoznačné, srozumitelné i uživatelům, je vhodné použít pojmů z daného oboru.
- Složitost jednoho DFD - formát na A4, přehlednost - dodržet pravidlo 7 (max. 7 uzlů).
- Formulace DFD - jasná a přehledná, z pohledu analytika i uživatele, návrháře.

# Pravidla pro tvorbu DFD

- Konzistentnost DFD - s ostatními modely (E-R diagram, DD, STD, ...).
- Kontrola správnosti po technické stránce, vyváženost po stránce estetické.
- Kontrola DFD mezi jednotlivými úrovněmi (stejně vstupní a výstupní datové toky, ukončení rozkladu minispecifikacemi).

# Pravidla pro tvorbu DFD

## Pravidla pro datové toky.

- Perpetuum mobile – proces jen s výstupními datovými toky.
- Černá díra – proces, který má jen vstupní datové toky a žádné výstupní.
- Datové paměti mohou být propojeny jen pomocí nějakého procesu.
- Datový tok z/do terminálu musí procházet přes proces.
- Datový tok mezi procesy znázorňují jen přenášená data, nikoli volání funkcí.



# Pravidla pro tvorbu DFD

Datový tok s tímž názvem může být v systému vícekrát, pokud znamená tentýž datový tok s tímtéž obsahem.

Pro pojmenování datových toků platí:

- jestliže tok z paměti přenáší celý výskyt dat, nemusí se pojmenovávat,
- přenáší-li se jeden nebo více celých záznamů, pojmenovává se stejně jako datová paměť,
- přenáší-li se pouze část záznamů, pojmenuje se jednoznačně jinak.

# Minispecifikace

Popisují logiku každé z funkcí na nejnižší úrovni DFD v dané větvi.

- U některých modelů jsou zahrnuty do modelu DFD, někde jsou uváděny jako zvláštní model funkční analýzy.
- Pro ověření by bylo ideální použít formální jazyk, ale z důvodu komunikace s uživateli je vhodnější množina slov přirozeného jazyka.

# Minispecifikace

Pravidla pro tvorbu minispecifikace:

- Každému elementárnímu procesu z poslední úrovně DFD odpovídá jedna minispecifikace.
- Minispecifikace popisuje postup, jak jsou vstupní data transformována na výstupní datové toky.

# Minispecifikace

- Minispecifikace popisuje věcně postupy a pravidla, kterými se transformace řídí, ne však konkrétní implementaci těchto pravidel.
- Je nutné hlídat možné redundance, které by mohly při tvorbě minispecifikací vzniknout.

# Minispecifikace - evidence výroby produktů

IF potvrzeni vyroby THEN

zapis do pameti SKLAD pocet vyrobenych kusu

IF prikaz k expedici THEN

SELECT:

CASE 1 (pocetKusuVeSKLADU >= pozadavekObchOddeleni):

vydej zboží

odepis ze skladu pocet kusu

CASE 2 (pocet kusu ve SKLADU < pozadavekObchOddeleni):

IF pocet kusu ve skladu je > 0 THEN

posli vse

nastav pocet kusu na 0

OTHERWISE

nic

zadost o dovyrobu

převzato z: <http://ananas.johny.cz/minispecifikace.htm>

# Vztah mezi modely DFD a ER

Výsledek funkční analýzy může ovlivnit již hotovou datovou analýzu.

Po nadefinování minispecifikací je nutno ověřit, zda existují všechny datové prvky.

- Existují všechny atributy?
- Existují tabulky pro uložení mezivýsledků?

Nutno zasáhnout do datové struktury a doplnit o další potřebné prvky.

# Zdroje

- <http://www.smartdraw.com/resources/tutorials/Data-Flow-Diagram-Notations>
- Structured Analysis and Systems Specification (DeMarco, 1978)
- [http://www.yourdon.com/jesa/pdf/JESA\\_obqujljp.pdf](http://www.yourdon.com/jesa/pdf/JESA_obqujljp.pdf)
- <http://www.fi.muni.cz/~xprist/PB007/?page=0#cile>
- <http://ananas.johny.cz/minispecifikace.htm>