

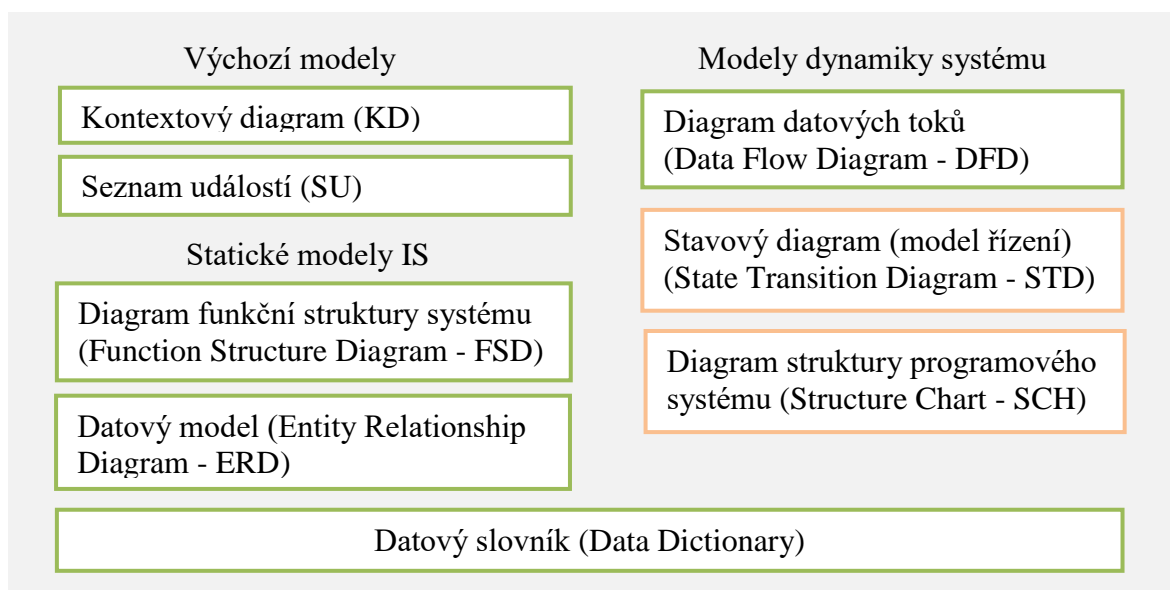
<b>Předmět:</b>	<b>Projektování IS</b>
<b>Téma:</b>	<b>Strukturovaný přístup k vývoji IS (část C)</b>

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

Obsah:

1. Stavový digram (STD).....	2
2. Diagram struktury programového systému (SCH) .....	5
3. Rekapitulace prezentovaných modelů IS .....	9
4. Historický přehled strukturovaných metod vývoje IS .....	10
5. Yourdonova strukturovaná analýza .....	14
6. Závěr k tématu .....	16

Opakování - „Modely strukturovaného přístupu ...“



## 1. Stavový digram (STD)

Účel modelu:

- zobrazit časově závislé **chování systému** (subsystému),
- zachytit **události a reakce** na ně,
- zachytit možné **stavy systému** a průchod těmito stavy.

Zobrazuje situace typu:

- stav 1 --> událost (podmínka, akce) --> stav 2.

Hlavní komponenty:

- uzly – znázorňují stavy,
- orientované hrany – přechody mezi stavy,
- skladiště událostí (Event Store).

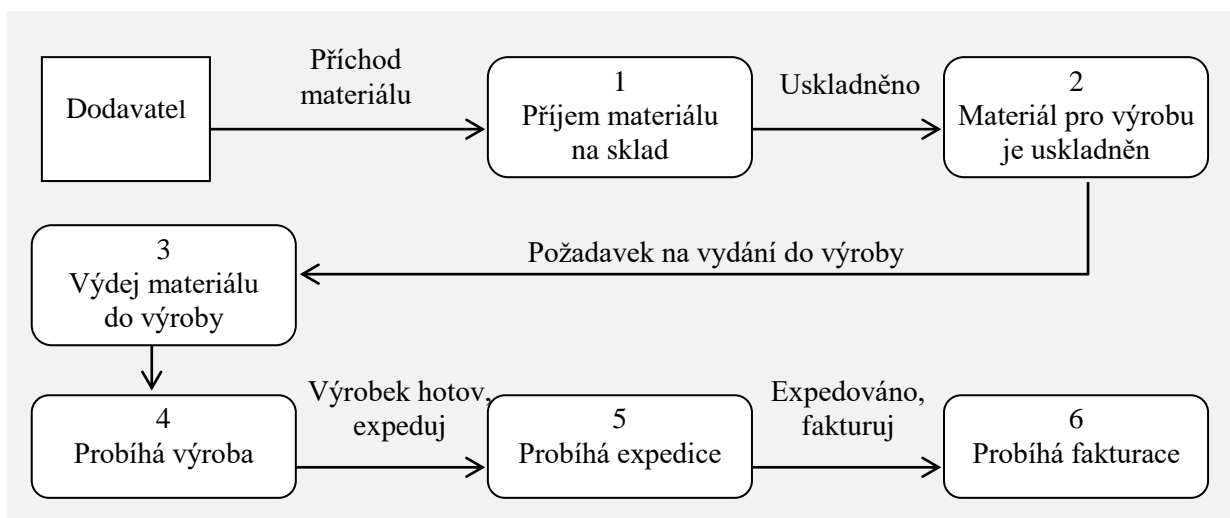
Čím je určen **stav systému**?

- parametry (tzv. stavové veličiny),
- rozmezí hodnot parametrů (charakterizuje daný stav).

Událost:

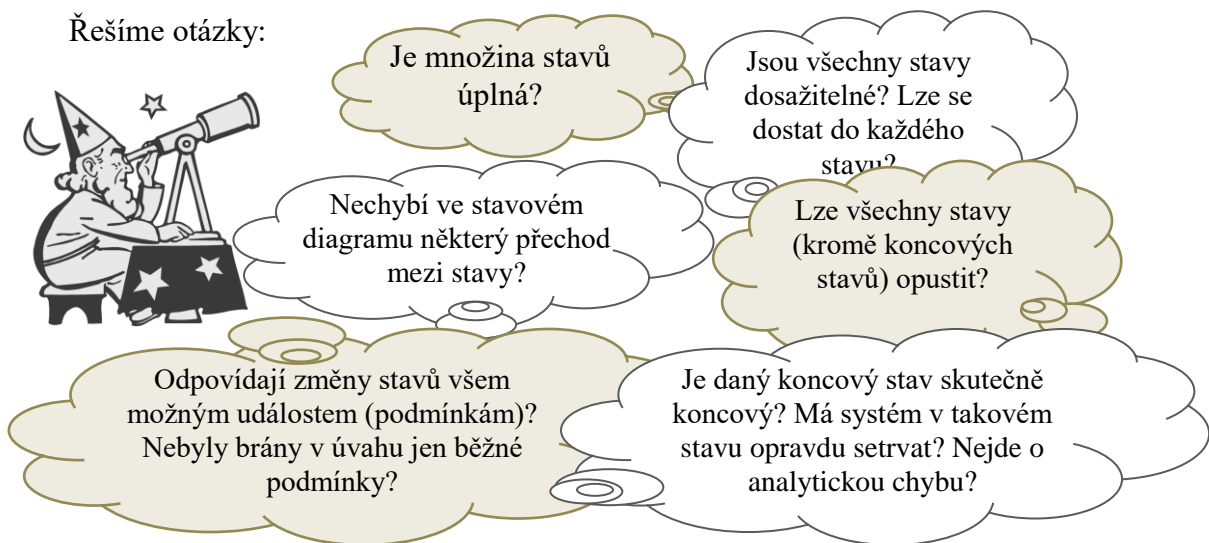
- jev, který vyvolává přechod mezi stavy,
- např. "uplynul stanovený čas", "přišla očekávaná data", ...
- srovnej např. F,T,C-události v modelu SU.

Grafické znaky - viz obrázek:



Postup tvorby stavového diagramu:

- 1) identifikujeme všechny stavy systému, které budeme rozlišovat,
- 2) identifikujeme všechny možné přechody mezi stavy a jejich příčiny,
- 3) určíme stav počáteční a stav (stavy) koncový,
- 4) ověření úplnosti modelu
  - o vyjdeme z počátečního stavu, zkoumáme přípustné změny stavu, přechody mezi stavy.



Hierarchie stavových diagramů:

- od nejhrubší rozlišovací úrovně -> potřebné detaily,
- dekompozice (rozklad) stavu na **podstavy**,
- stavy sekvenční (následují po sobě) a paralelní (souběžné).

Využití STD:

- modelování systémů (subsystémů) s **real-time** charakteristikami,
- rozpracování logiky **řídícího procesu** v DFD,
- modelování stavů standardních a chybových (!),
- modelování stavů zpracování dat dle provozních technologií.

Příklad – možné stavy provozních technologií:

- např. stavy procesu „zpracování poštovních poukázek B“:
  - stav č. 1 – příjem předpisů plateb od hromadných podavatelů,
  - stav č. 2 – zpracování předpisů plateb, tisk a distribuce Pk B,
  - stav č. 3 – load dat podaných Pk do centrální databáze a jejich opravy,
  - stav č. 4 – load dat vyplacených Pk z pošt a jejich opravy,
  - stav č. 5 – párování podaných Pk na vyplacené Pk,
  - stav č. 6 – řešení nevyplacených Pk a vad,
  - stav č. 7 – vyúčtování s podavatelem Pk,
  - stav č. 8 – řešení reklamací a druhopisů,
  - ... ,
- obdobně lze rozlišit:
  - stavy účetního systému během účetního období apod.

Z procesního a technického pohledu je důležité:

- ošetřit také **chybové stavy** systému,
- ošetřit stavy **nouzového provozu** systému (např. na náhradním uzlu v clusteru, v záložní lokalitě apod.).

Vztah mezi STD a ostatními modely:

- STD jako návazný diagram na Seznam událostí (SU),
- STD jako rozpracování řídicího procesu DFD,
- souvislost s Data Dictionary (slovní popis prvků).

Poznámka – souvislost s modely OOP:

- v OOP máme tzv. State diagram (zobrazení stavů objektů, subsystémů),
- v OOP pro modelování real-time systémů – navíc Timing diagram.

## 2. Diagram struktury programového systému (SCH)

Kde se nyní nacházíme?

- na **implementační úrovni** zobrazení vyvíjeného IS,
- v etapě detailního návrhu systému.



V této etapě máme již k dispozici:

- KD (SU) - model začlenění systému do okolní infrastruktury IS/IT,
- DFD včetně detailů - elementární procesy a jejich minispecifikace,
- ERD - model fyzických datových struktur (strukturu dat) systému,
- případně STD.

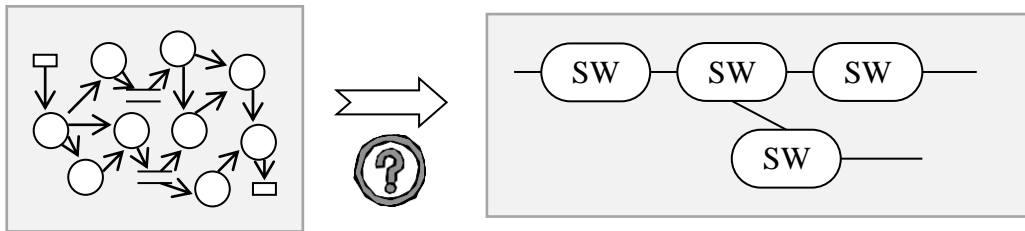
Co je v této etapě tvorby IS je nutné?

- navrhnout podrobnou fyzickou (HW a SW) **architekturu systému**,
- přiřadit jednotlivé části (subsystémy) vyvíjeného IS hardwarovým a **softwarovým komponentám**,
- navrhnout vhodnou fyzickou reprezentaci dat,
- navrhnout (rekapitulovat) uživatelské rozhraní systému,
- navrhnout technicky zajištění požadavků na bezpečnost a dostupnost dat,

- zpracovat časový a věcný plán pro etapu implementace,
- zpracovat plán testování systému,
- zpracovat plán školení uživatelů,
- zpracovat plán procesu zavádění IS do provozu.

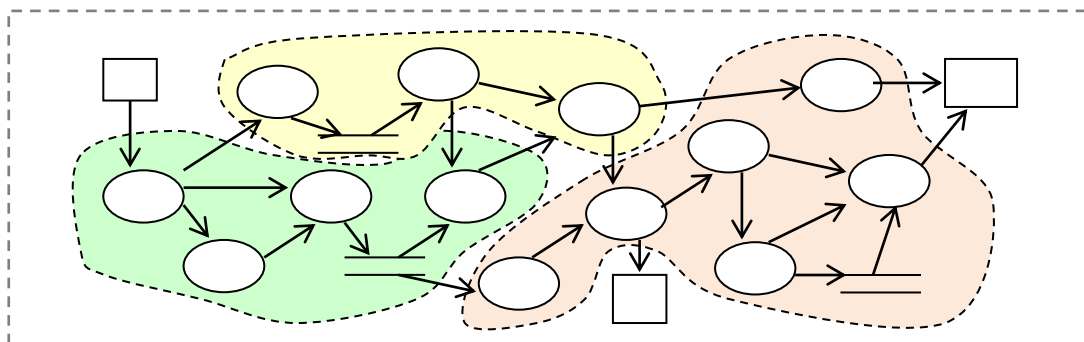
Účel modelu SCH (Structure Chart):

- znázornění **hierarchie SW modulů**,
- vazeb mezi SW moduly.



Tvorba struktury programového systému zahrnuje:

- definovat pokrytí funkcí systému SW komponentami (programovými moduly),
- identifikovat vhodný rozsah jednotlivých SW modulů,
- identifikovat závislosti mezi moduly a rozhraní jednotlivých SW modulů,
- vyčlenit SW sub-moduly, používané (volané) na více místech systému,
- integrovat SW moduly v jeden celek – výsledný IS.



Spolupráce rolí při tvorbě SCH:

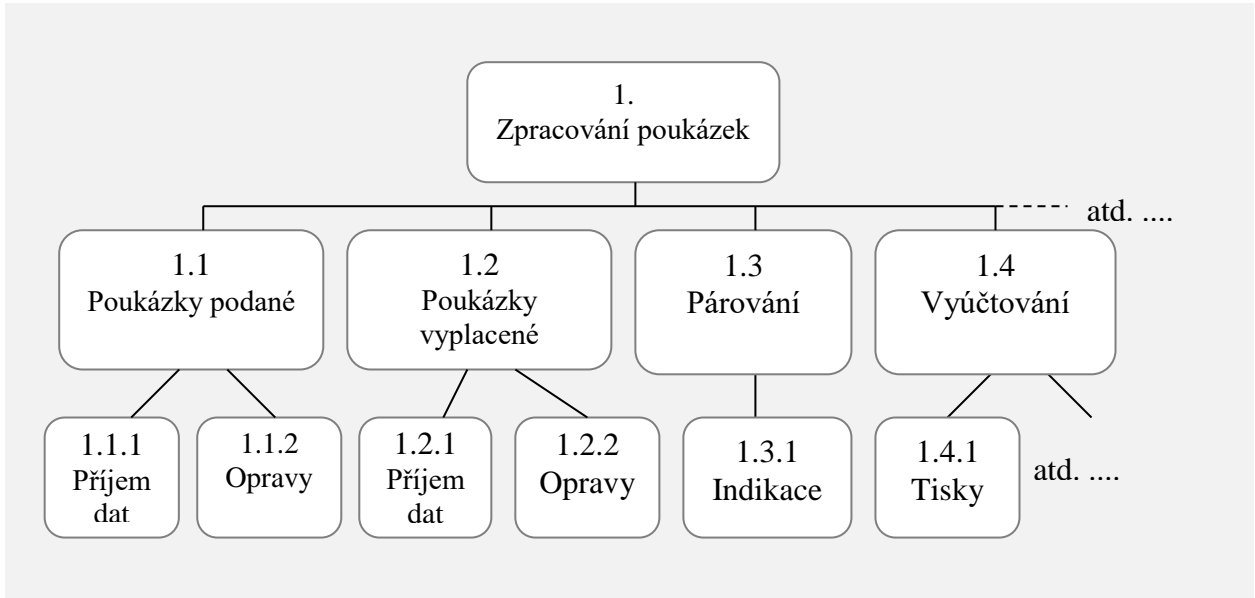
- analytik a programátor.

Řešení otázek:

- optimální velikosti SW modulů,
- globálních proměnných a funkcí,
- možnosti opakovaného využití SW modulů (procedury).

Grafické znaky - viz obrázek:

- SW moduly, volání modulů, předávané hodnoty parametrů, větvení, cyklická volání, ...



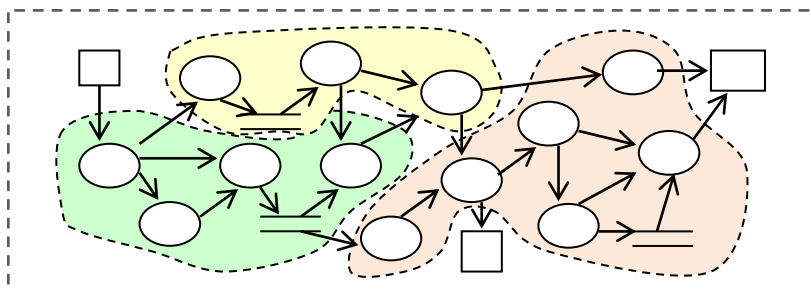
Obrázek: Ukázka diagramu struktury programového systému - SCH

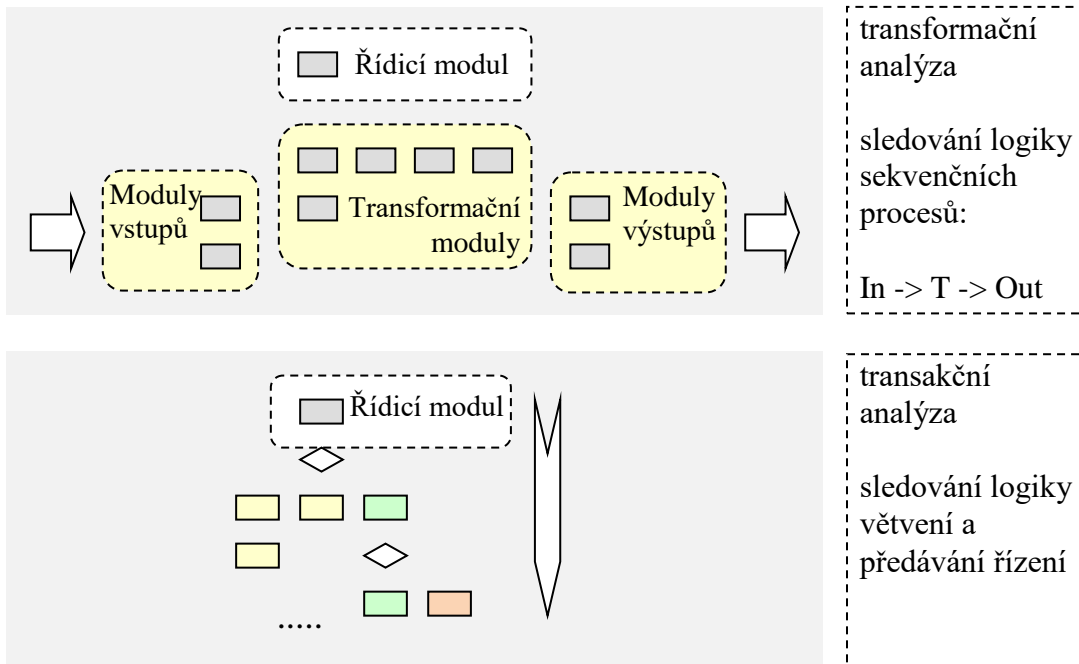
Poznámka:

- může vzniknout nejen stromová struktura, nýbrž také síť SW modulů.

Techniky tvorby SCH:

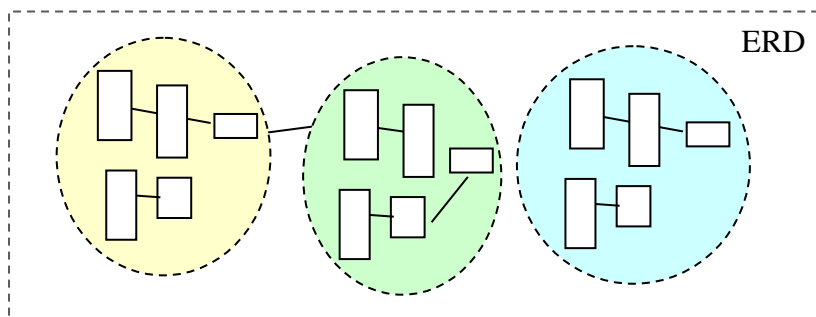
- modulární návrh na základě **funkční dekompozice**
  - vycházíme z FSD,
- modulární návrh na základě **analýzy datových toků**
  - vycházíme z DFD,
  - technika transformační analýzy,
  - technika transakční analýzy,





- modulární návrh na základě [analýzy datových struktur](#)

- vycházíme z ERD,
- uvažujeme skupiny procesů, které pracují s „příbuznými“ typy dat,
- tyto skupiny procesů sdružujeme do SW modulů.



Souvislost s jinými modely:

- viz výše odstavec Techniky tvorby SCH,
- kontrola konzistence: DFD (FSD, ERD) vs. SCH.

Poznámka – souvislost s modely OOP:

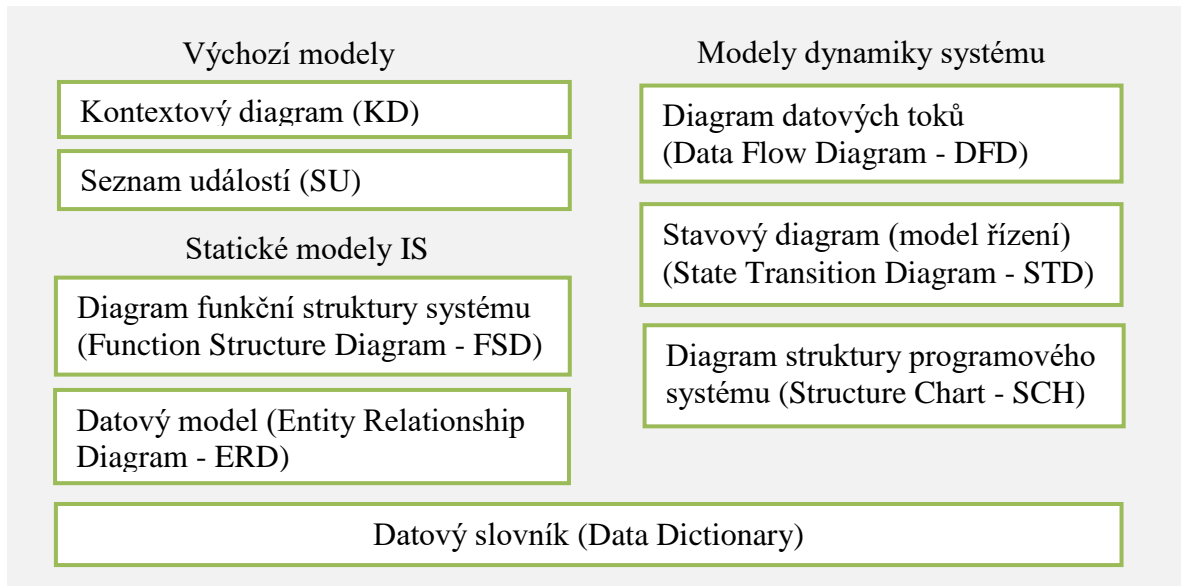
- Diagram komponent (tj. SW komponent),
- Diagram nasazení (Deployment diagram).



### 3. Rekapitulace prezentovaných modelů IS

Seznámili jsme se:

- s hlavními myšlenkami (principy) strukturované analýzy a návrhu IS,
- s modely používanými k zobrazení IS.



Poznámka:

- Nechybí nám (z hlediska současné praxe) nějaký model?
- Pokud myslíte, že chybí, pak proč asi chybí?
- A jak si bez onoho „chybějícího modelu“ v praxi pomůžete?

Co je nutné při modelování IS pochopit?

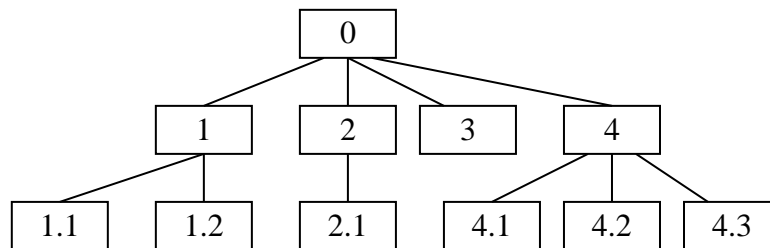
- **účel** použitých modelů
  - co pomocí daného modelu mohu (chci) zobrazit,
  - k jakému účelu bude vytvářený model IS sloužit,
  - komu (jakému čtenáři) je výsledek mé práce určen, ...
- **úplnost** modelu
  - zda v něm něco důležitého nechybí,
  - zejména na koncepční a technologické úrovni zobrazení,
- zvolit vhodnou (dostatečnou) **úroveň podrobnosti** zobrazení (detailů),
  - v závislosti na účelu modelu a na cílovém čtenáři,
  - podrobnost (úplnost) vs. srozumitelnost a vypovídací schopnost,



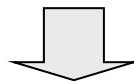
- poznat „informační potřebu“ cílového čtenáře (!!),
- dobře zkontrolovat **konzistenci** modelů
  - bezspornost různých modelů,
  - bezspornost rozlišovacích úrovní téhož typu modelu.
- celkově **zhodnotit** návrh (model) IS
  - Odpovídá návrh IS specifikovaným požadavkům?
  - Bude navržený systém skutečně řešit požadované úkoly?
  - Bude splňovat požadavky zákazníka (uživatele)? Bude kvalitně uspokojovat jeho potřeby?

#### 4. Historický přehled strukturovaných metod vývoje IS

a) Metoda HIPO (Hierarchy Input-Process-Output)



Obrázek: Hierarchie SW modulů

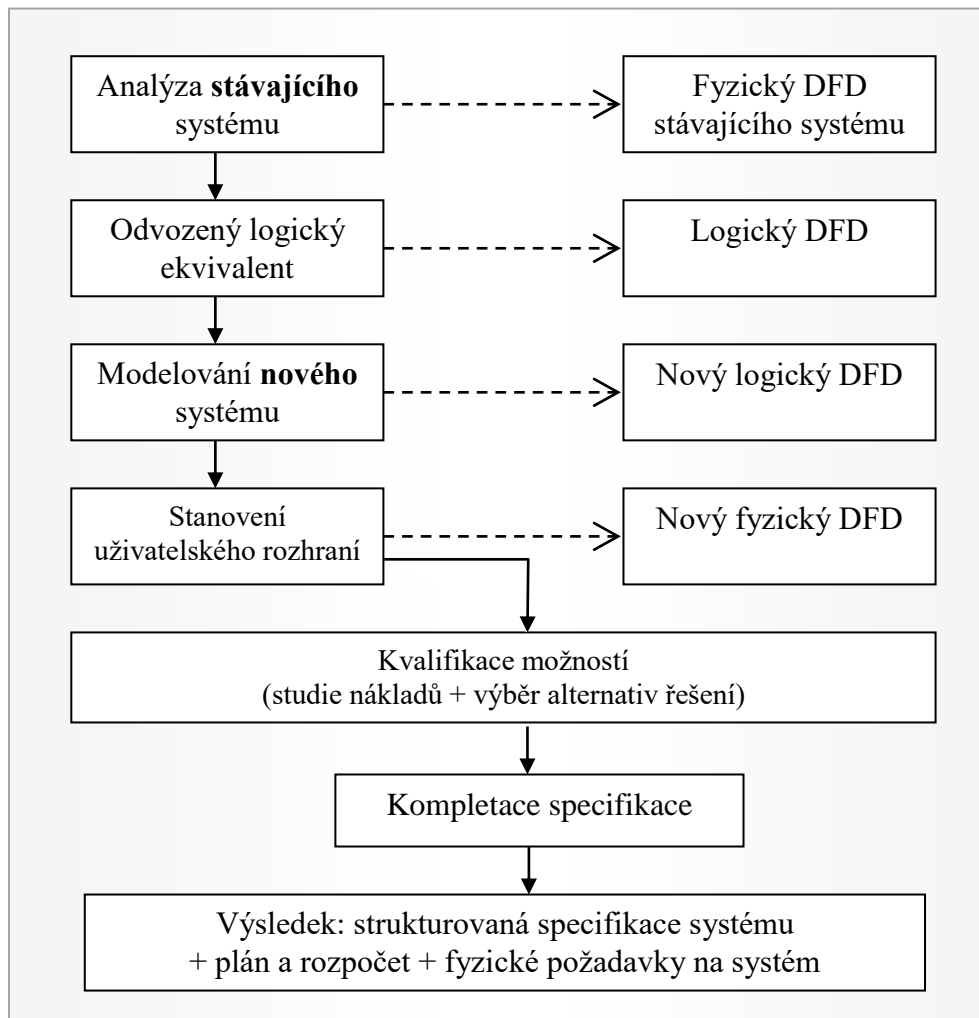


Autor: J.Novák Diagram: 2.1	Program: Zpracování PkB Modul: Příjem PkB	Datum: 20.10.2000 Strana: 5 z 15
--------------------------------	--	-------------------------------------

Input	Proces	Output
Soubor PkB podavatele  Číselník podavatelů	Čtení souboru PkB Uložení do pracovní tabulky Kontroly podacích znaků .... .... Čtení kontrolního součtu Kč .... Uložení do tabulky přijatých Pk	Načtená data PkB  Soubor chyb

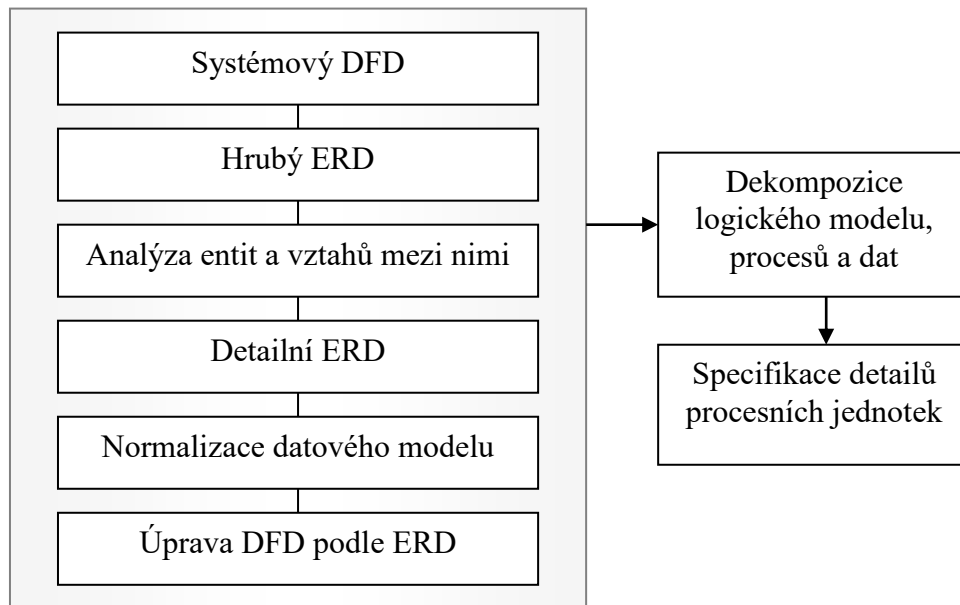
Obrázek: IPO diagram pro dílčí proces 2.1

b) Metoda DeMarco



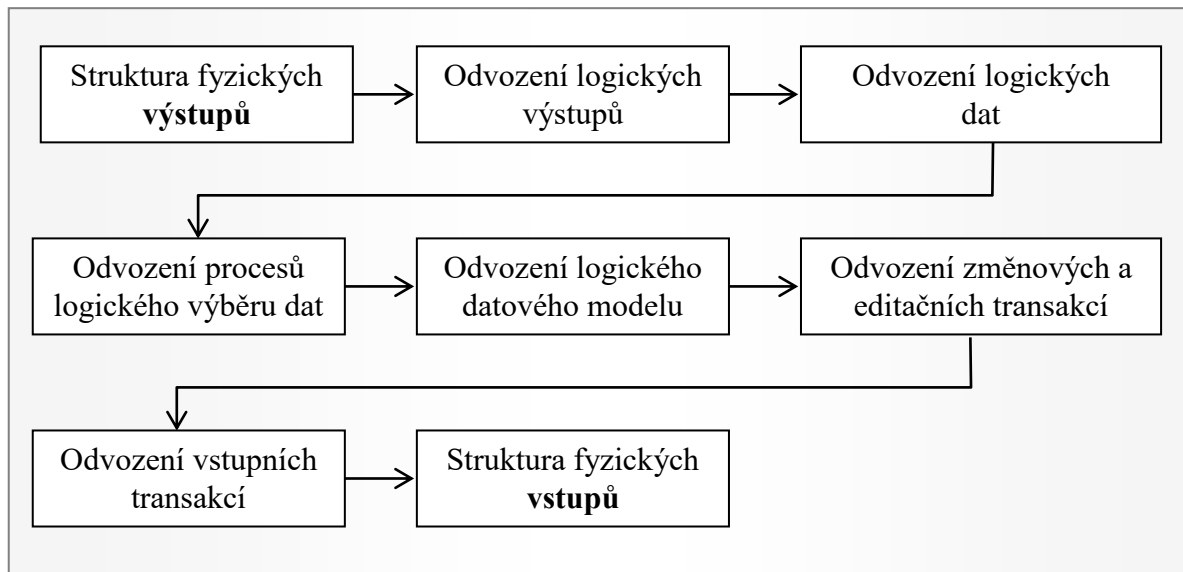
Obrázek: Schéma metody DeMarco

c) Metoda Gane/Sarson

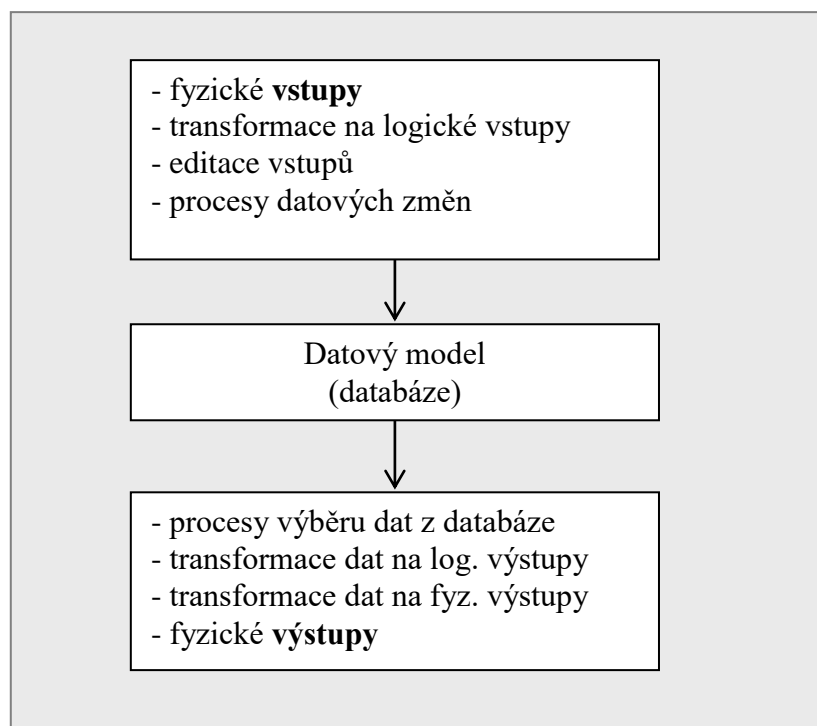


Obrázek: Schéma metody Gane/Sarson

d) Metoda Warnier/Orr



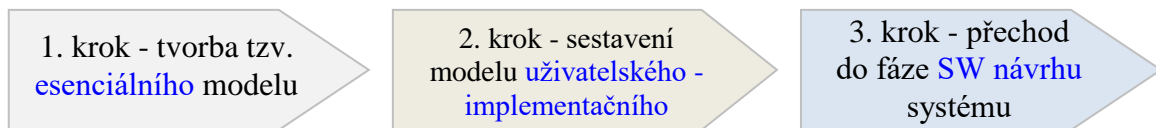
Obrázek: Schéma metody Warnier/Orr



Obrázek: Struktura informací o vyvíjeném IS

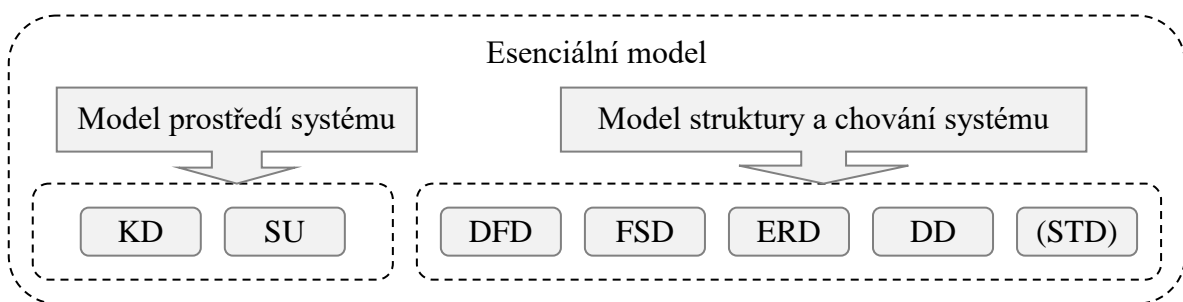
## 5. Yourdonova strukturovaná analýza

Doporučený postup:



Esenciální model:



- esence = podstata,
- esenciální model = model, zobrazující podstatu systému.



Počáteční modely IS:

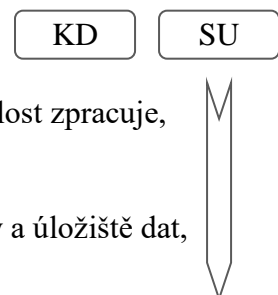
- seznam událostí,
- kontextový diagram.

Otázka řešená v praxi:

- Na základě jakých faktorů je vhodné provádět **dekompozici DFD**? 
- Yourdonova metoda doporučuje:
  - proveďte dekompozici DFD pomocí techniky **analýzy událostí**,
  - tj. použijte **seznam událostí (SU)** jako inspiraci k dekompozici DFD. 

Technika analýzy událostí:

- 1) vytvoř **seznam událostí (SU)**,
- 2) pro každou událost ze SU vytvoř proces v DFD, který událost zpracuje,
- 3) procesy pojmenuj podle odezvy na událost,
- 4) k procesům urči vstupní datové toky, výstupní datové toky a úložiště dat,
- ...

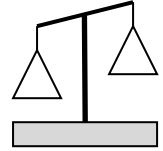


- 5) výsledný DFD porovnej s KD (předpoklad – máme KD),
- 6) ověř bezspornost a úplnost modelu.

DFD

Vyvažování DFD (level balancing):

- cíl - získat DFD na různých rozlišovacích úrovních,
- vyvažování DFD směrem nahoru,
- vyvažování DFD směrem dolů.



Předpoklad techniky analýzy SU:

- předpokládáme existenci SU (vycházíme z SU),
- Ale co když nemáme SU vytvořený?
- doporučení - vytvoř SU pomocí KD nebo hrubého ERD.



První postup tvorby SU - pomocí KD:

- vytvoříme KD (terminátory + datové toky s okolím),
- zkoumáme terminátory a toky dat z okolí,
- odhalujeme události (toky dat ze strany externích entit), na které systém musí reagovat,
- => získáváme SU.

KD



SU

Druhý postup tvorby SU - pomocí ERD:

- vytvoříme prvotní „hrubý“ datový model (ERD),
- pro každou množinu entit hledáme vnější událost, která vede k jejímu použití („Co hýbe s daty?“),
- => takto sestavíme SU,
- => na základě SU dále sestavíme KD.

ERD

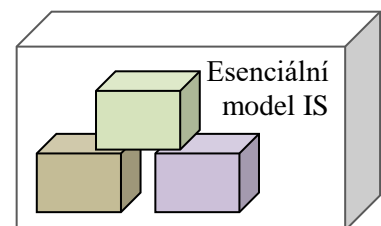


SU

KD

Výsledek tvorby tzv. **esenciálního modelu**:

- kontextový diagram (KD),
- seznam událostí (SU),
- model datových toků (DFD),
- model datových struktur (ERD),



- slovní popis v datovém slovníku (DD),
- případně stavový diagram (STD).

Kontroly konzistence modelů:

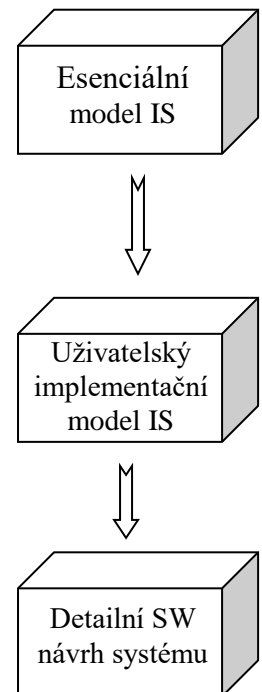
- konzistence mezi různými modely,
- konzistence mezi rozlišovacími úrovněmi téhož modelu,
- DFD a datový slovník,
- DFD a minispecifikace (Process Specification),
- DFD a ERD,
- DFD a STD.

Uživatelský **implementační model**:

- návrh uživatelského rozhraní (pro vstupy a výstupy dat),
- automatizované a neautomatizované činnosti, integrace těchto činností,
- specifikace hodnot a množství dat (frekvence přístupů, přístupové klíče, ....),
- řešení bezpečnosti a spolehlivosti systému,
- ....

Přechod do fáze **SW návrhu** systému:

- návrh struktury programového systému,
- viz téma Structure Chart (SCH).



## 6. Závěr k tématu

Seznámili jsme se:

- s **hlavními myšlenkami** (principy) strukturované analýzy a návrhu IS,
- s **modely** používanými k zobrazení IS,
- s příklady historicky významných **strukturovaných metod** vývoje IS,
- se základními myšlenkami **Yourdonovy** strukturované analýzy.



## Rekapitulace pohledů na vyvíjen IS:

