

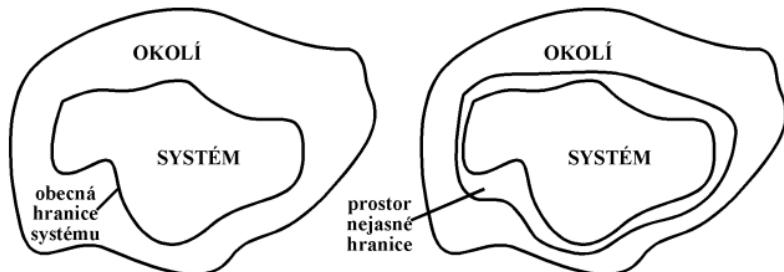
7 251202 - DFD a IS

Hranice mezi systémem a okolím

- Rozhraní mezi sys a okolním světem popisuje **model okolí**.
- Vnitřní části sys popisuje **model chování**.
- Úkol analýzy - stanovit, co je a co není součástí systému.
 - Yourdonova Moderní strukturovaná analýza, 1989 (YMSA):
 - Esenciální model = model okolí + model chování

Volba hranice systému

- Analytik může ovlivnit volbu hranice v prostoru nejasné hranice.



- Nebezpečí: Malý rozsah projektu identifikuje pouze symptomy, velký rozsah projektu je obtížně zvládnutelný, nebo neuspěje vůbec.

Definice okolí - Nástroje

- Účel systému
 - krátký, stručný textový dokument
 - pro pracovníky vrcholového řízení
 - Vyjadřuje strategické cíle, kterých by mělo být dosaženo po realizaci a nasazení systému.
- Kontextový diagram
 - zvláštním případem DFD
 - Obsahuje jediný proces, který reprezentuje celý systém. Zdůrazňuje tyto rysy systému:
 - terminátory - lidé a systémy z okolí, s nimiž systém komunikuje
 - data přijímaná z vnějšího světa, která mají být nějak zpracována
 - data produkovaná, které systém zasílá do vnějšího světa
 - datové paměti sdílené systémem a terminátory
 - hranice mezi systémem a ostatním světem
- Seznam událostí
 - textový výčet stimulů

- objevují se ve vnějším světě a na něž musí systém odpovědět.

Příklad kontextového diagramu



Události

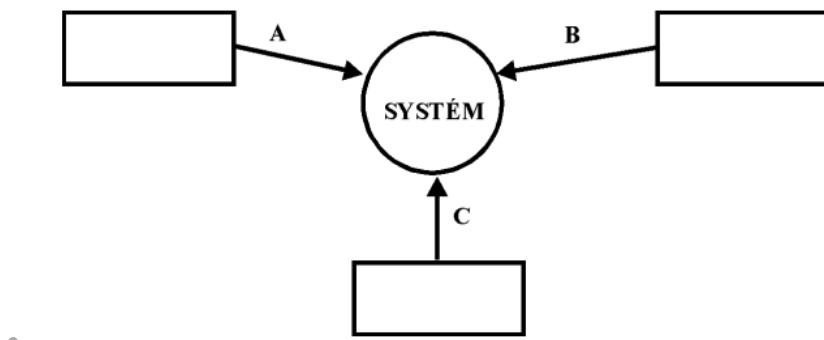
- = Seznam událostí
 - textový výčet stimulů, které se objevují ve vnějším světě a na něž musí systém odpovědět.
 - Každá událost je označena F, T nebo C.
- F (Flow)
 - tokově orientovaná událost
 - sdružená s datovým tokem (příjem jednoho nebo několika datových paketů)
- T (Temporal)
 - časová událost
 - nastává v nějakém významném časovém bodě (absolutní nebo relativní čas)
- C (Control)
 - řídící událost, povel, signál
 - je sdružena s vnějším řídícím tokem, povelem

Příklady FTC

1. U1: Zákazník vystavuje objednávku. (F)
2. U2: Zákazník ruší objednávku. (F)
3. U3: Vedení požaduje zprávu o prodeji každé ráno. (T)
4. U4: Pokyn k vyhodnocení stavu a vývoje prodeje. (C)

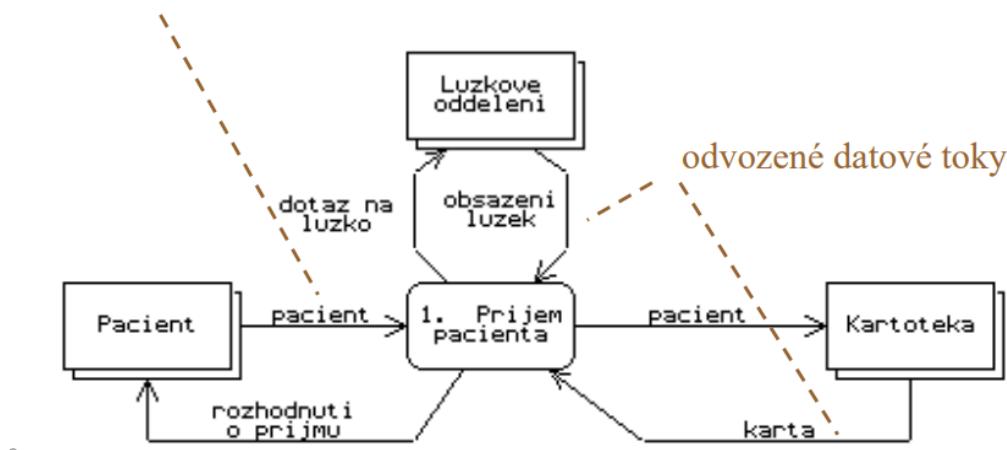
Vztahy mezi datovými událostmi

- Ne všechny vnější datové toky jsou sdruženy s F-událostmi.
- A - tok spojený s událostí
- B, C - toky explicitně vyžádané pro řešení události



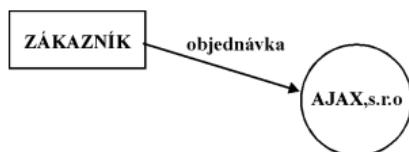
Příklad DFD - Ambulantní příjem

- F-událost: Příchod pacienta



Sestavení modelu okolí

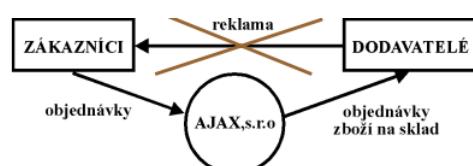
- Komunikace znázorněná na kontextovém diagramu:



a) přímá komunikace terminátor - proces



b) komunikace přes externí paměť

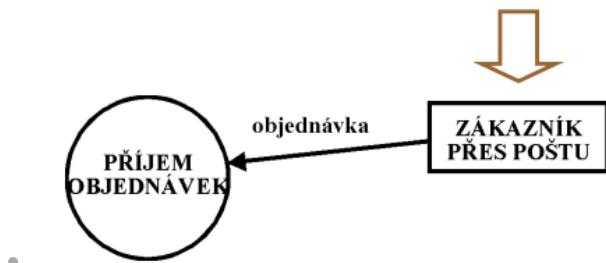


c) komunikace terminátor - terminátor: chybný kontextový diagram

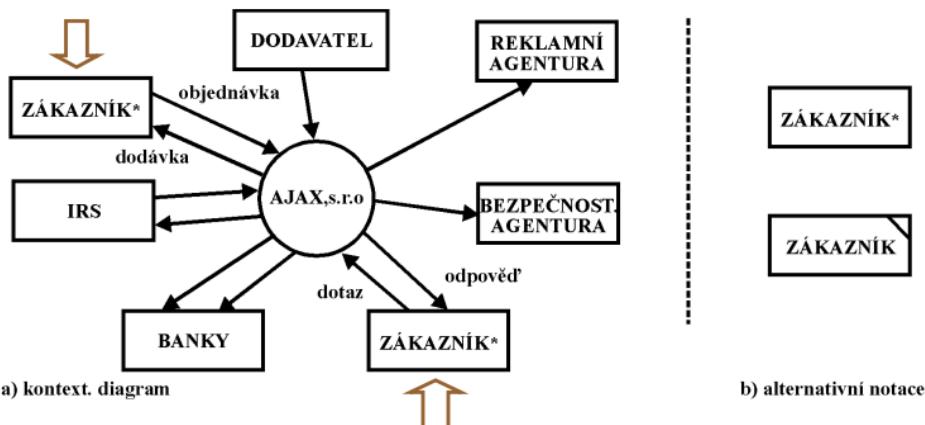
Poznámky k terminátorům

- Terminátory s velkým počtem vstupů/výstupů – kvůli přehlednosti zakreslíme vícekrát.
- Terminátor je individuální osoba (subjekt) - je třeba zjistit roli vzhledem k systému a pojmenovat terminátor podle této role.

- Rozlišíme mezi zdroji dat a nosiči (nositeli) dat. Přednost dáme pojmenování podle zdroje dat, nosič dat je „technologický prvek“, při realizaci může být použito odlišné „medium“.
 - Kompromisní pojmenování terminátoru: „zdroj i nosič dat“ v jednom jméně.



Duplikované terminátry



Dialogové toky

- Pokud systém posílá data až po výzvě, je tato výzva esenciální a musí být zakreslena na kontext.



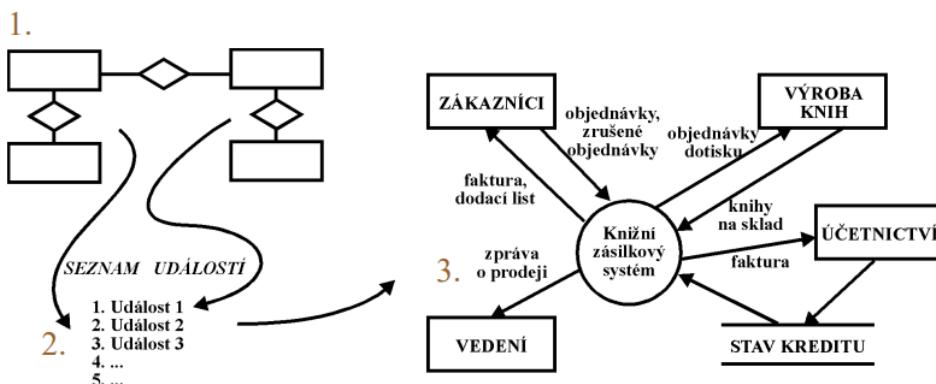
- Dvojice odpovídajících vstup-výstupních toků lze někdy nahradit obousměrným tokem.



Pořadí tvorby modelů okolí

- Varianty možných postupů:
 - Na základě informací od uživatelů lze sestavit kontextový diagram. Zkoumáním terminátorů a toků odhadneme události. Prověříme pomocí dalších modelů (procesní, datový m.).
 - Začneme s datovým modelem - ERD. Nalezneme esenciální objekty a jejich vztahy. Pro každou entitu hledáme, jaké vnější události vedou k jejímu použití, změně atd. Sestavíme předběžný seznam událostí. Pomocí něho vytvoříme kontextový diagram.
 - Kontextový diagram => Seznam událostí => ...
 - ERD => Seznam událostí => Kontextový diagram => ...

Identifikace událostí pomocí ERD



Odhadnuté události:

- U1: Zákazník objednává knihu.
U2: Zákazník ruší objednávku.
U3: ...

pravděpodobný terminátor
- Zákazník

Dokončení modelu okolí

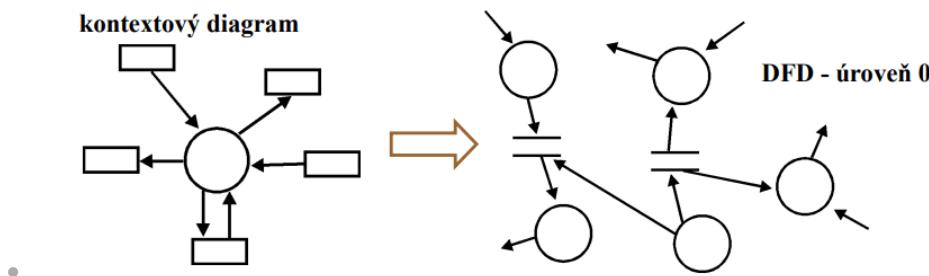
- Model okolí je nutné prověřit:
 - Každý vstupní tok na kontext. diagramu je nezbytný pro rozpoznání události, nebo pro vytvoření odezvy na událost, nebo obojí současně.
 - Každý výstupní tok je odezvou na událost.
 - Každá nečasová událost ze seznamu událostí by měla mít vstup podle něhož systém detekuje její výskyt.

- Každá událost musí produkovat okamžitý výstup jako odezvu, nebo by měla uložit data pro pozdější výstup, nebo by měla změnit stav systému.

Vytvoření prvotního modelu chování

- Výchozí podklady - model okolí systému, který obsahuje dokumenty:
 - kontextový diagram,
 - seznam událostí,
 - dokument o účelu systému,
 - zahájena tvorba datového slovníku
- Tvoříme model chování systému, tj. dekompozici systému na jednotlivé procesy, toky a paměti uspořádané v sadě DFD a doplněné o příslušné mini specifikace a definici datových komponent.

Kritika postupu „shora dolů“



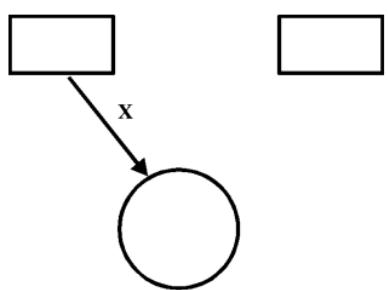
- Ochrnutí analýzy
 - Analytik nedokáže nalézt dekompozici na hlavní subsystémy.
- Jev 6 analytiků
 - Systém obsahuje právě tolik subsystémů, kolik je analytiků.
- Náhodné fyzické členění
 - Subsystémy tvořeny podle existujícího fyzického a organizačního členění.

Princip tvorby modelu chování

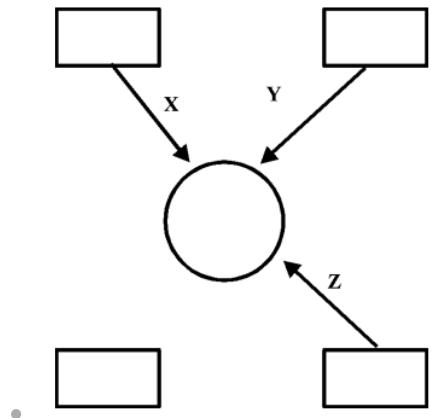
- Pro každou odezvu na událost ze seznamu událostí zakreslíme do prvotního DFD jeden proces.
- Proces pojmenujeme podle očekávané odezvy na tuto událost.
- Zakreslíme datové paměti, které modelují data nezbytná pro zpracování asynchronně probíhajících událostí.
- Doplníme odpovídající vstupní a výstupní toky.
- Vytvořený DFD ověříme proti kontextovému diagramu.

Události a průvodní toky

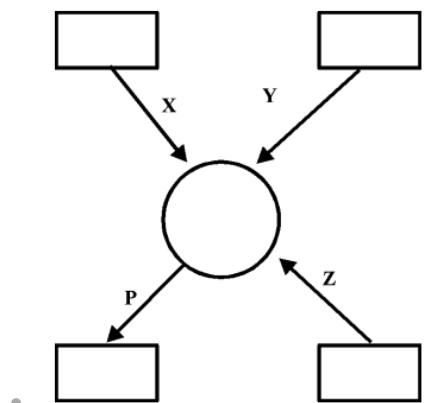
1. Výskyt události (datový tok X)



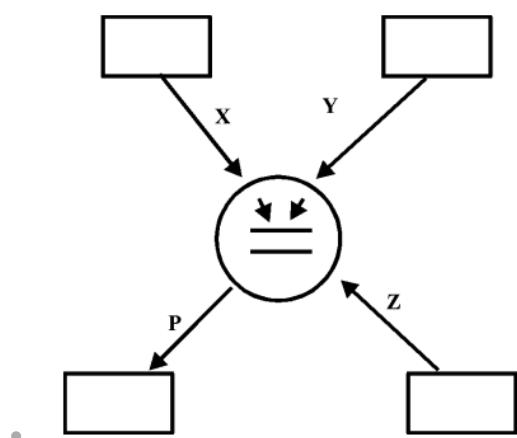
2. pro řešení události požadovány další údaje z okolí systému (Y,Z)



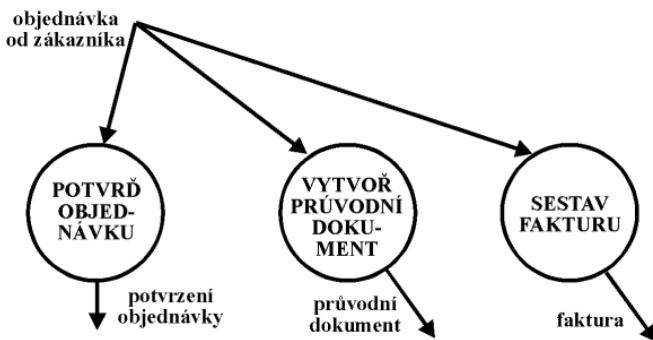
3. odezva na událost ve formě výstupního toku (P)



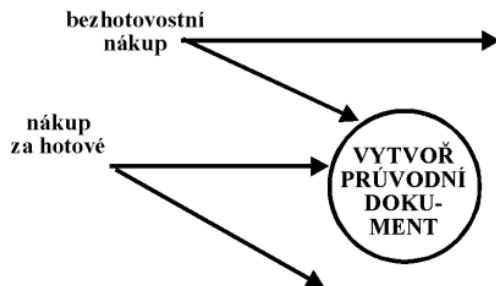
4. odezva na událost ve formě uložený



Události s více odezvami



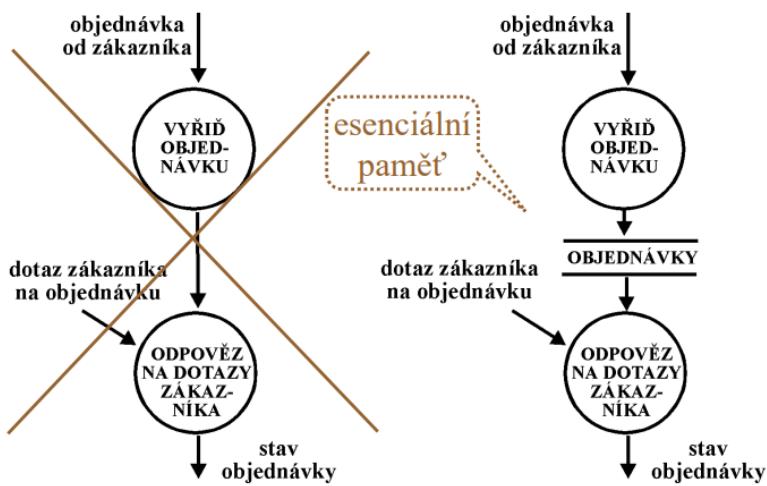
- Všechny odezvy (generující procesy) používají stejný vstupní tok.
- Všechny odezvy jsou vzájemně nezávislé.



- Odezva produkovaná procesem musí být identická pro různé události.
- Vstupní a výstupní data jsou identická pro různé události.

Propojení odezv na události

- Odezva na událost může vyžadovat data vytvořená jinou událostí.
- V esenciálním modelu pracuje každý proces nekonečně rychle.
 - Každý datový tok představuje komunikační cestu s nulovým dopravním zpožděním.
- Události, které na sobě závisí, můžeme synchronizovat jedině pomocí paměti.
 - Paměť je esenciální, je vynucena časováním vně systému.



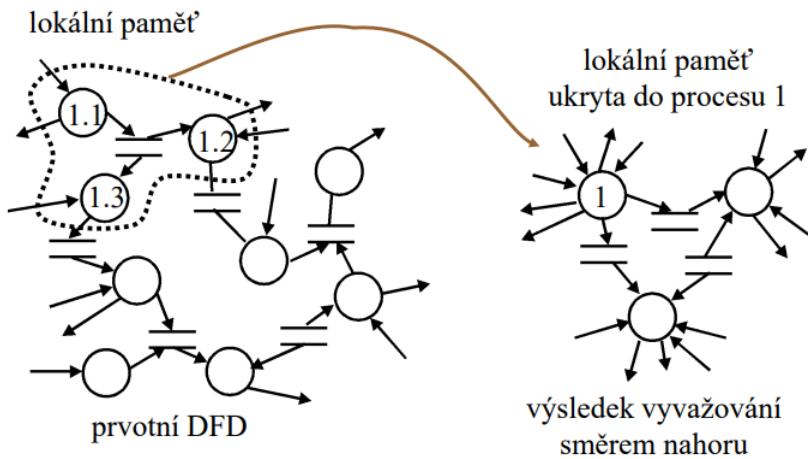
Prvotní model chování

- prvotní DFD <=> prvotní ERD
- paměti <= => entity

- Kontrola:
 - Zpracovává prvotní DFD každou událost?
 - Jsou zakresleny všechny potřebné vstupy a výstupy pro každou u
 - Jsou zakreslena nezbytná propojení mezi událostmi?
- Prvotní model chování nekonzultujeme s uživatelem ! Model není uspořádaný tak, aby mohl být pochopen jako celek.

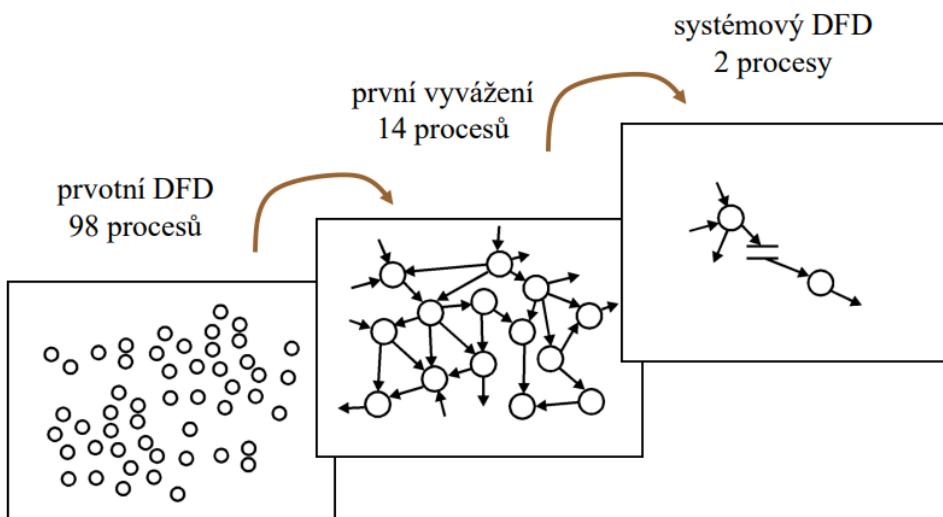
Vyvažování směrem nahoru

- Hledáme seskupení vzájemně souvisejících procesů do agregovaného procesu na diagramu vyšší úrovni.
1. Seskupení procesů má zahrnout blízké, obdobné odpovědi (obdobně pojmenované procesy).
 2. Paměti zakrýváme, pokud paměť používá pouze skupina procesů na nižší úrovni (žádný jiný proces vně této skupiny paměť nepoužívá).



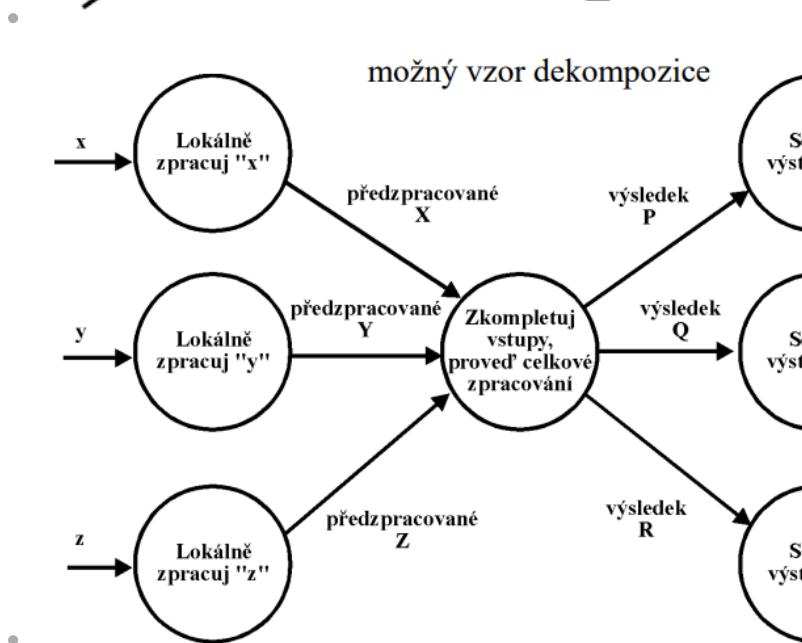
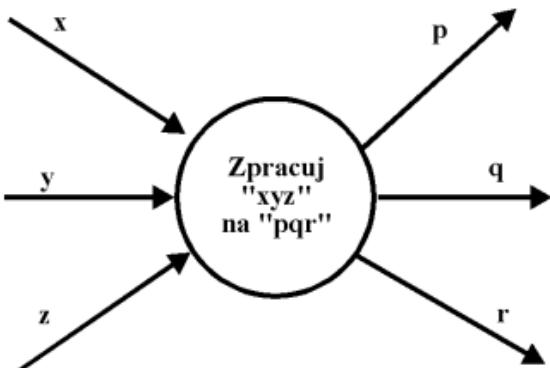
- Ukrývání paměti má přednost před pravidlem „7±2 procesů na jednom DFD“

Vícenásobné vyvažování směrem nahoru



Vyvažování směrem dolů

- Čistá funkční dekompozice
 - u procesu se složitou funkcí jsou zřetelné dílčí funkce.
 - Tyto funkce vyjádříme jako procesy na nižší úrovni DFD.
- Funkční dekompozice nezřetelná
 - Pokusíme se odvodit dekompozici na základě vstupních a výstupních toků.
- Data určují směr dekompozice.



Dokončení modelu chování

- Dokončení ERD a datového modelu jako celku:
 - ERD vyvíjen obdobně jako DFD a souběžně s DFD.
 - prvotní ERD
 - přiřazování atributů k entitám
 - identifikace nových nebo přebytečných entit
 - křížová kontrola proti DFD
- Dokončení stavového modelu:
 - Stavové diagramy plní úlohu minispecifikace řídících procesů.
 - definice všech možných stavů
 - dosažitelnost všech stavů

- cesta z nekoncových stavů
- reakce na všechny možné podmínky v každém stavu