E.Yourdon

Strukturovaná analýza

Hranice mezi systémem a okolím

Rozhraní mezi systémem a okolním světem popisuje ***model okolí*****.**

Vnitřní části systému popisuje ***model chování*****.**

Úkol analýzy - stanovit, co je a co není součástí systému.

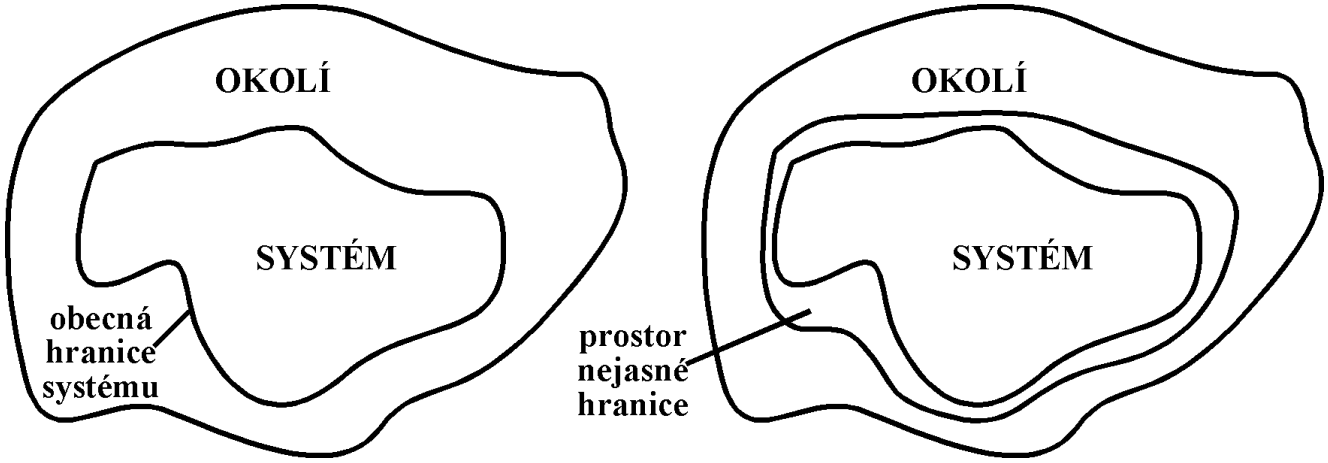
Yourdonova Moderní strukturovaná analýza, 1989 (YMSA):

Esenciální model = model okolí + model chování

Volba hranice systému

Hranice mezi světem a systémem je náhodná a může být stanovena:

- na základě manažerského rozhodnutí
- jako výsledek politického kompromisu
- zcela náhodně



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

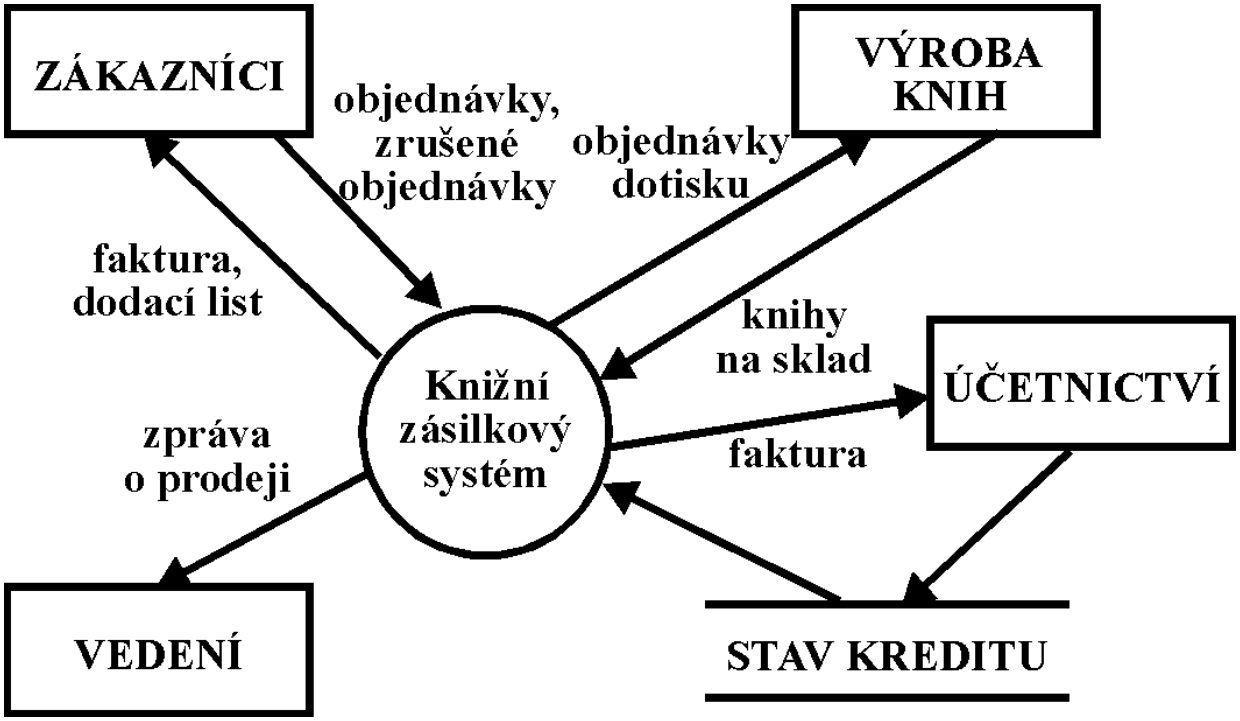
Účel systému je krátký, stručný textový dokument, určený především pro pracovníky vrcholového řízení. Vyjadřuje strategické cíle, kterých by mělo být dosaženo po realizaci a nasazení systému.

Kontextový diagram je 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 je textový výčet stimulů, které se objevují ve vnějším světě a na něž musí systém odpovědět.

Příklad kontextového diagramu



Jaký je „účel“ systému který je v souladu s tímto kontextovým diagramem?

Události

Seznam událostí je textový výčet stimulů, které se objevují ve vnějším světě a na něž musí systém odpovědět.
Každá událost je označena F, T nebo C.

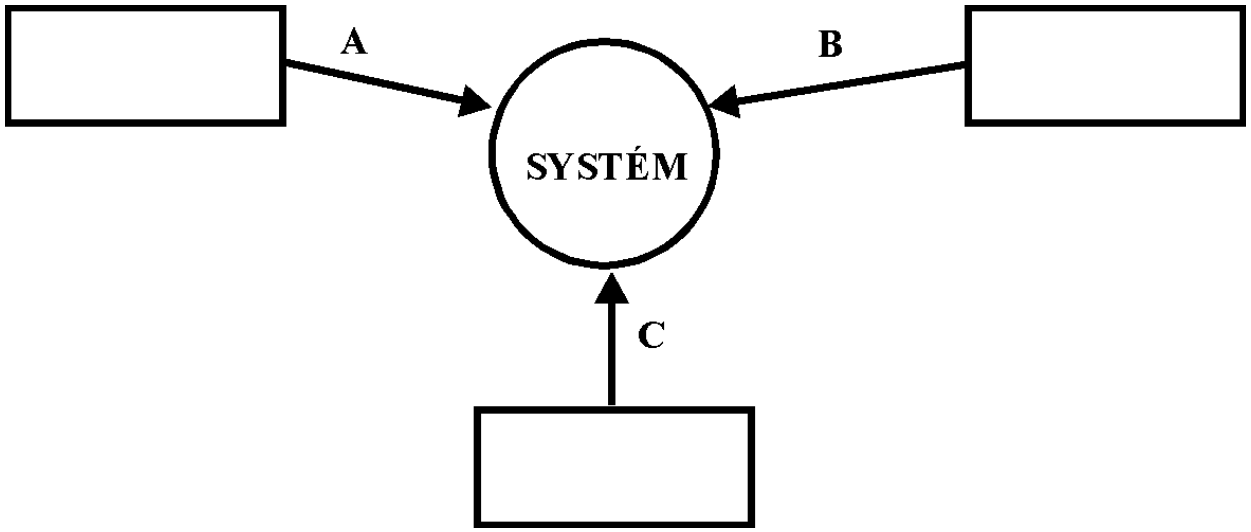
F (Flow): tokově orientovaná událost, je sdružená s datovým tokem (příjem jednoho nebo několika datových paketů)
T (Temporal): časová událost, nastává v nějakém významném časovém bodě (absolutní nebo relativní čas)
C (Control): řídící událost, povel, signál (je sdružena s vnějším řídícím tokem, povelem)

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

Vztahy mezi datovými toky a událostmi

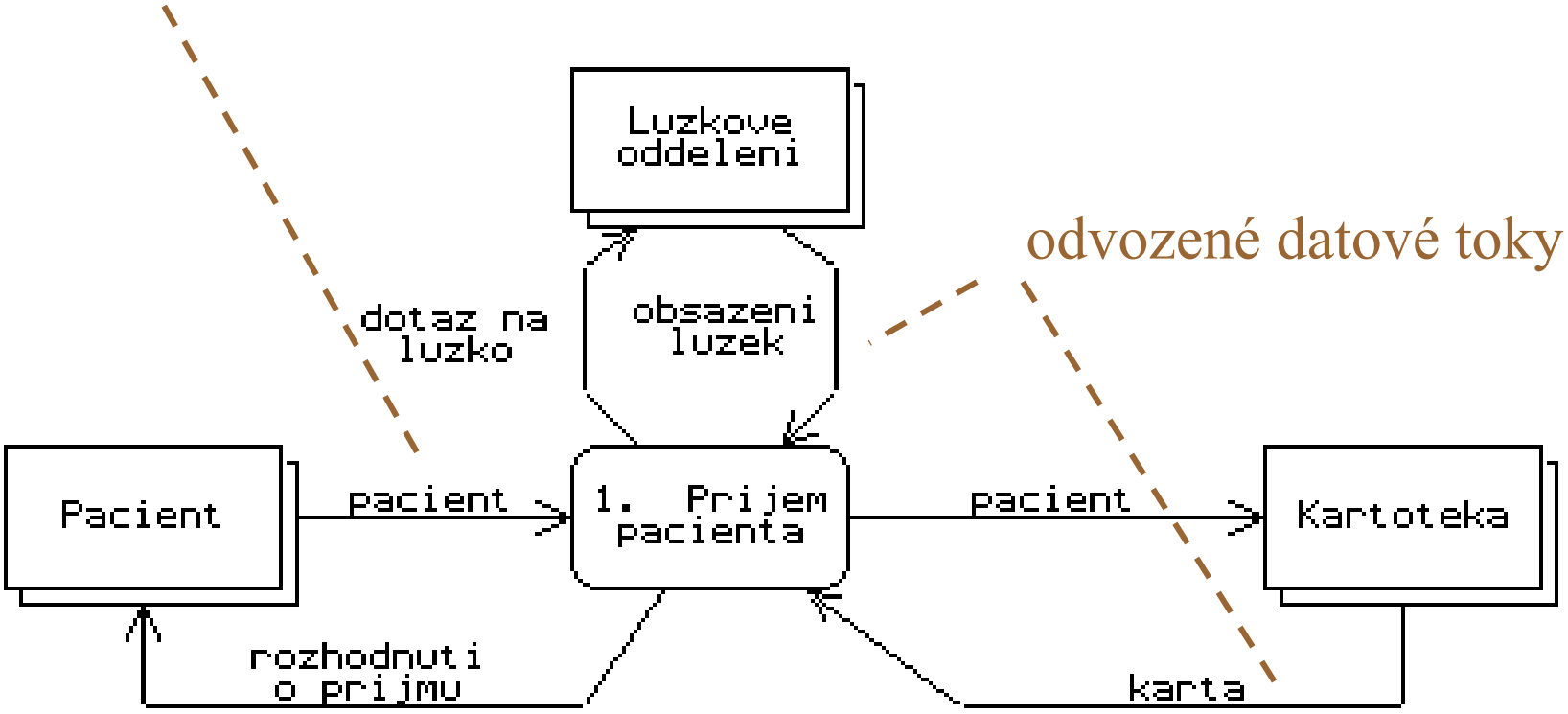
Ne všechny vnější datové toky jsou sdruženy s F-událostmi.

A - tok spojený s událostí
B, C - toky explicitně vyžádané pro řešení události



Příklad DFD – Ambulantní příjem

F-událost: Příchod pacienta

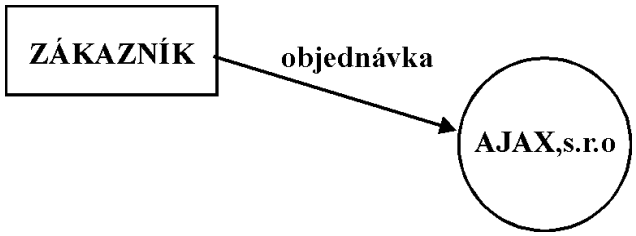


Proč to není tak jednoduché, jak se zdá ...

- Nikdo plně nerozumí rozsahu systému tak, jak byl původně definován. Mnozí uživatelé poskytují při rozhovorech pohled „z vnitra systému ven“.
- Uživatelé mají převážně lokální znalosti, znají pouze část systému. Pohledy lokálních uživatelů jsou vzájemně rozporné.
- Některé uživatele nebo jejich důležité interakce při analýze opomeneme. Opomenutí je považováno za chybu analýzy a tedy za chybu analytika.

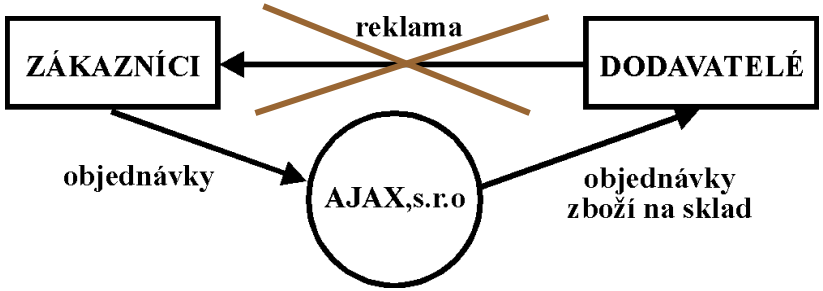
Sestavení modelu okolí

Komunikace znázorněná na kontextovém diagramu:



a) přímá komunikace terminátor - proces

b) komunikace přes externí paměť

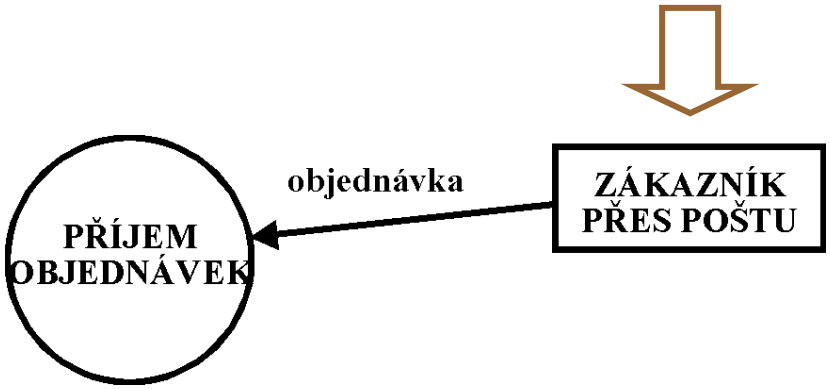


c) komunikace terminátor - terminátor: chybný kontextový diagram

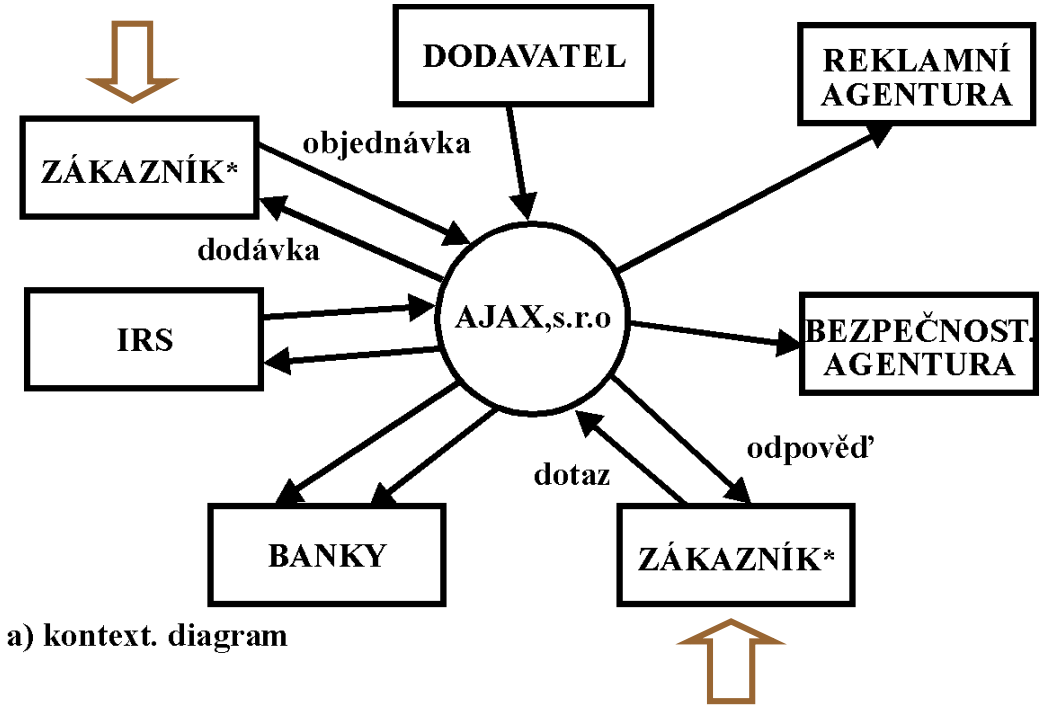
Poznámky k terminátorům

- 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 „médium“.

Kompromisní pojmenování terminátoru: „zdroj i nosič dat“ v jednom jméně.



Duplikované terminátory

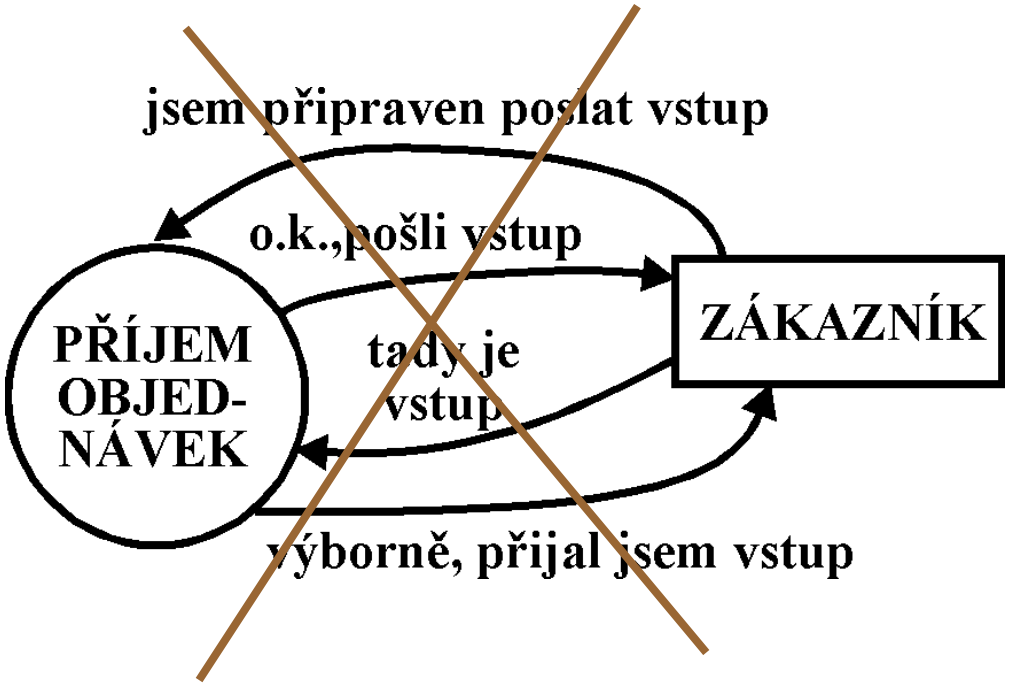


a) kontext. diagram

b) alternativní notace

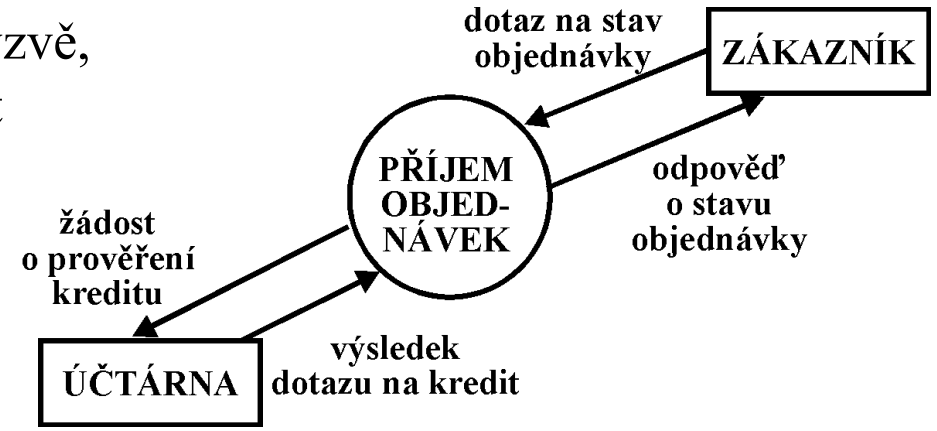
Dialogové toky

Tok (řídící tok) na kontextovém diagramu modeluje data (povely) přicházející do systému, předávaná(é) ven ze systému. U esenciálního modelu se při znázornění toků vyhýbáme implementačně závislým prvkům dialogu:

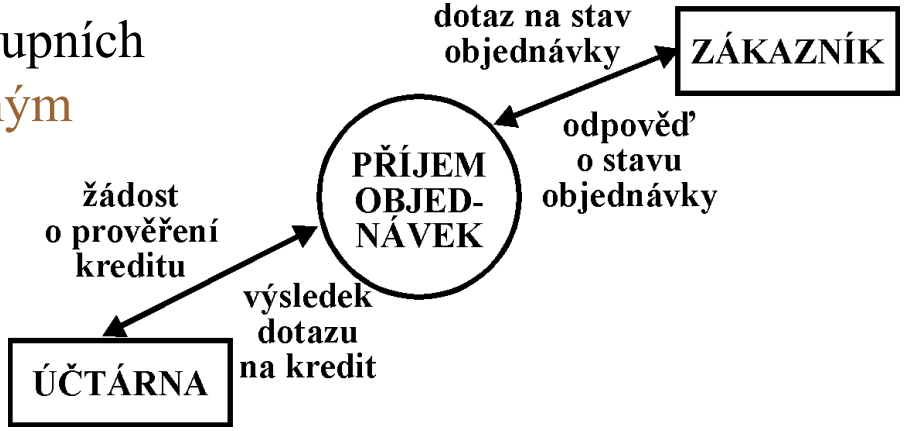


Dialogové toky

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



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



Seznam událostí - Poznámky

Událost nebo tok související s událostí?

Př.: a) Zákazníkova objednávka je přijata systémem.

b) Zákazník zaslal objednávku.

Význam správného pojmenování:

Nechceme, aby některé toky byly považovány za události, jsou-li to ve skutečnosti vyžádané údaje pro řešení jiné události.

Do seznamu událostí zařazujeme i chybové situace.

Zajímají nás pouze možné chyby na straně terminátorů, u esenciálního modelu neočekáváme vnitřní chyby systému.

Pořadí tvorby modelů okolí

Varianty možných postupů:

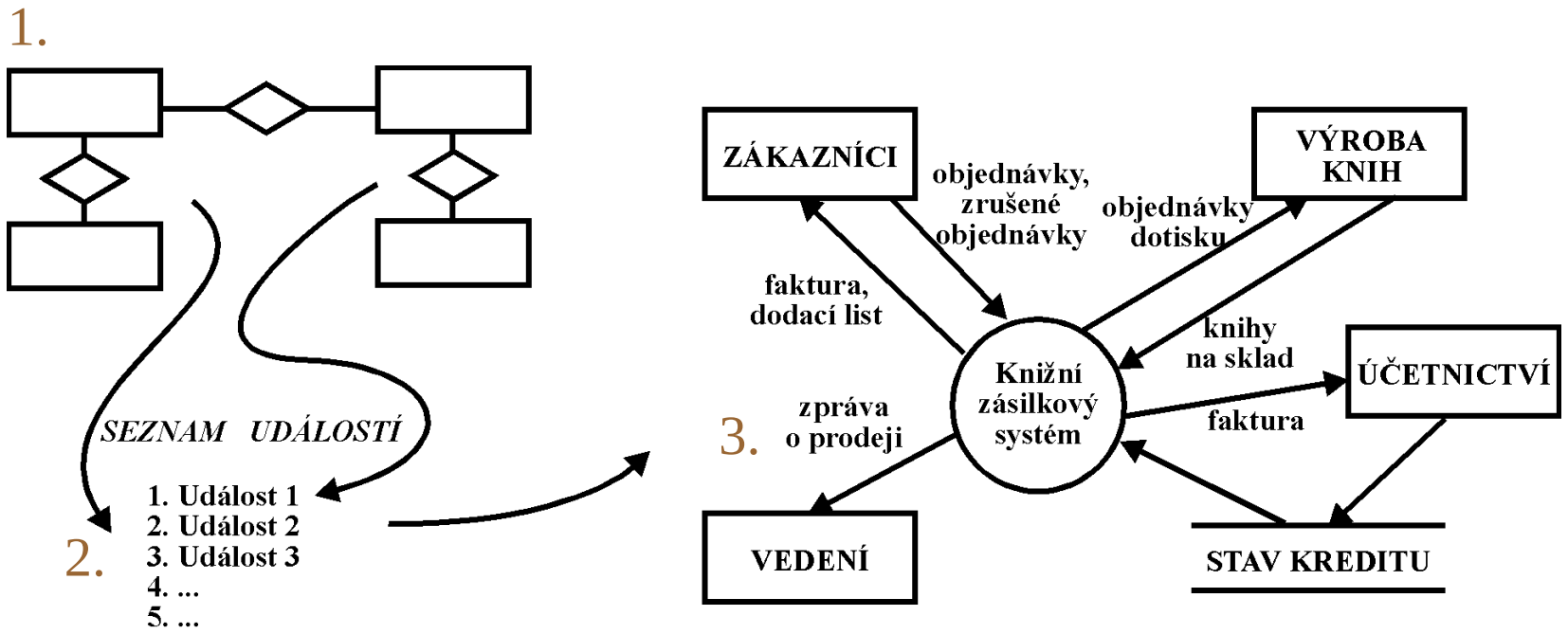
1. Na základě informací od uživatelů lze sestavit kontextový diagram. Zkoumáním terminátorů a toků odhadneme události. Prověříme pomocí dalších modelů (procesní, datový m.).

   Kontextový diagram => Seznam událostí => ...

2. 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.

   ERD => Seznam událostí => Kontextový diagram => ...

Identifikace událostí pomocí ERD



Odhadnuté události:

U1: Zákazník objednává knihu.

U2: Zákazník ruší objednávku.

U3: ...

⇒ pravděpodobný terminátor - Zákazník

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

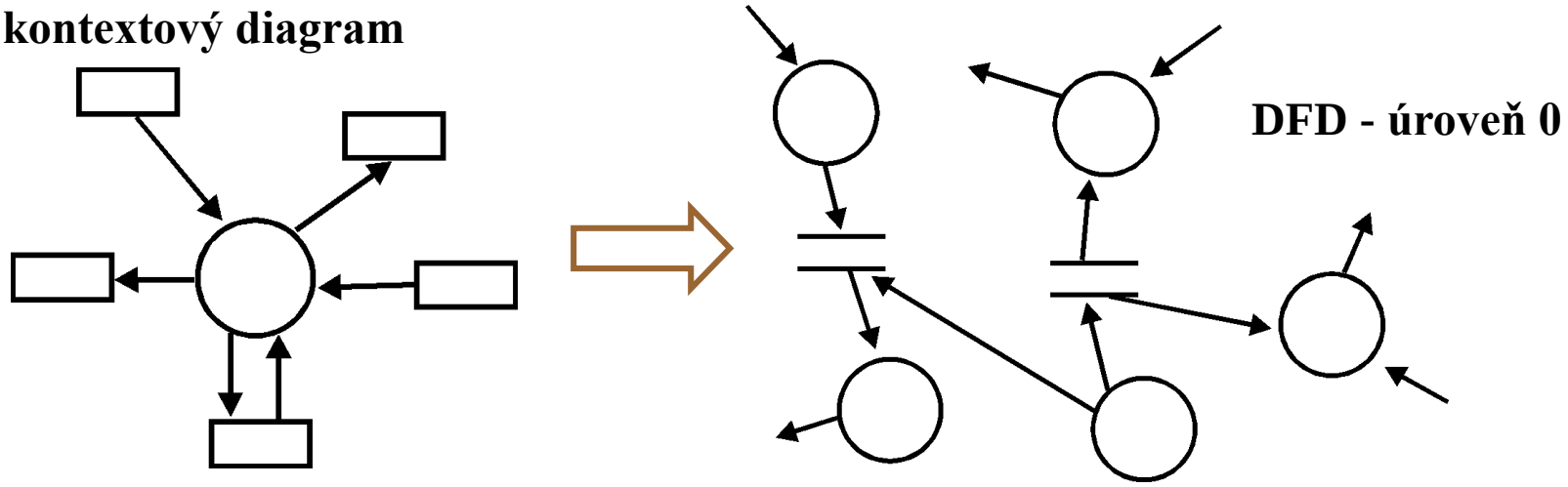
Výchozí podklady - *model okolí systému*, který obsahuje dokumenty:

- kontextový diagram,
- seznam událostí,
- dokument o účelu systému,
- zahájena tvorba datového slovníku (datová rozhraní mezi systémem a terminátory)

⇩

Tvoříme *model chování systému*, tj. dekompozici systému na jednotlivé procesy, toky a paměti uspořádané v sadě DFD a doplněné o příslušné minispecifikace a definici datových komponent.

Kritika postupu „shora dolů“



Ochrnutí analýzy: Analytik nedokáže nalézt dekompozici na hlavní subsystémy.

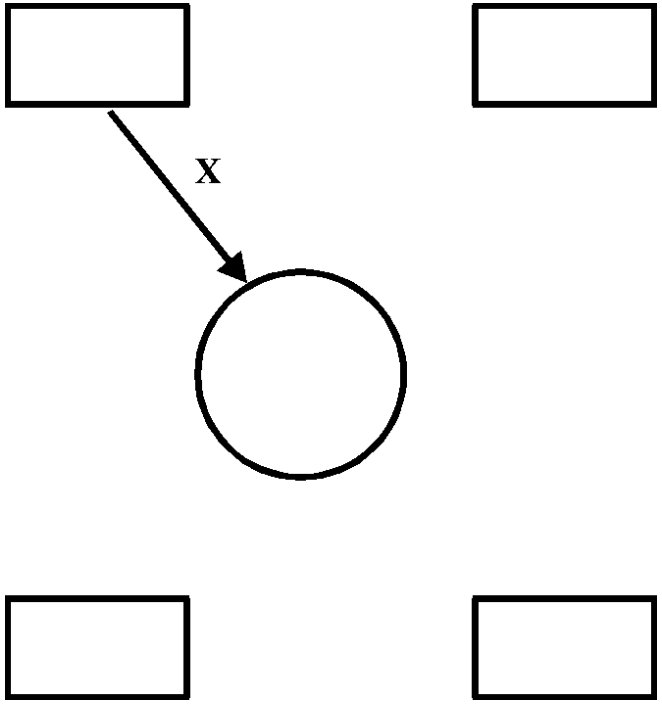
Jev 6 analytiků: Systém obsahuje právě tolik subsystémů, kolik je analytiků.

Náhodné fyzické členění: Subsystémy tvořeny podle existujícího fyzického a organizačního členění.

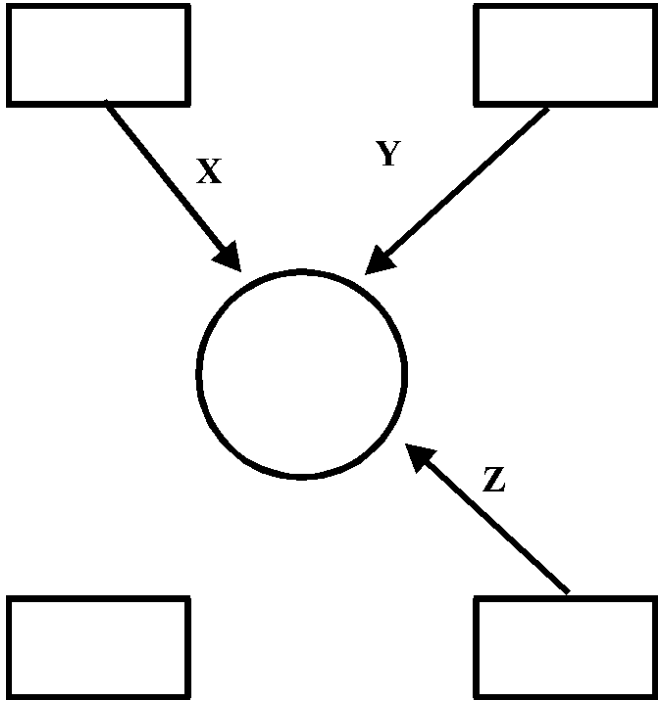
Princip tvorby modelu chování

- 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

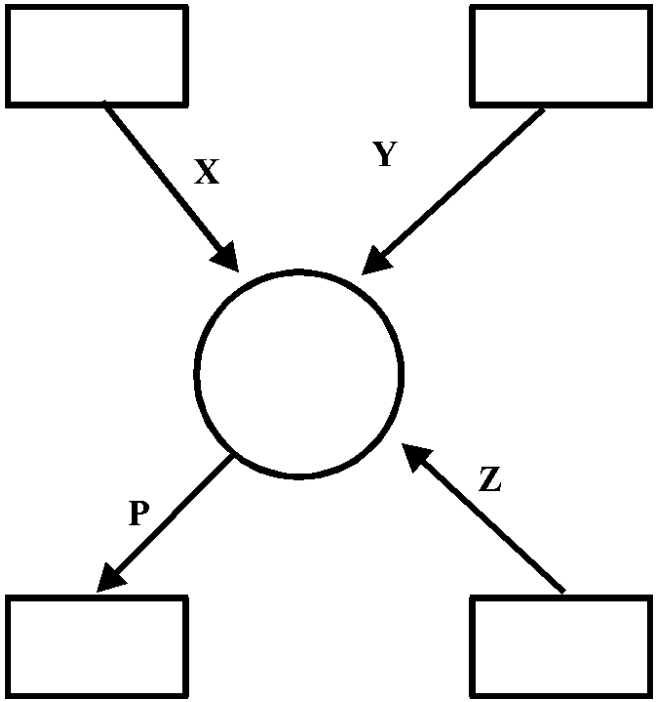


1) výskyt události
(datový tok X)

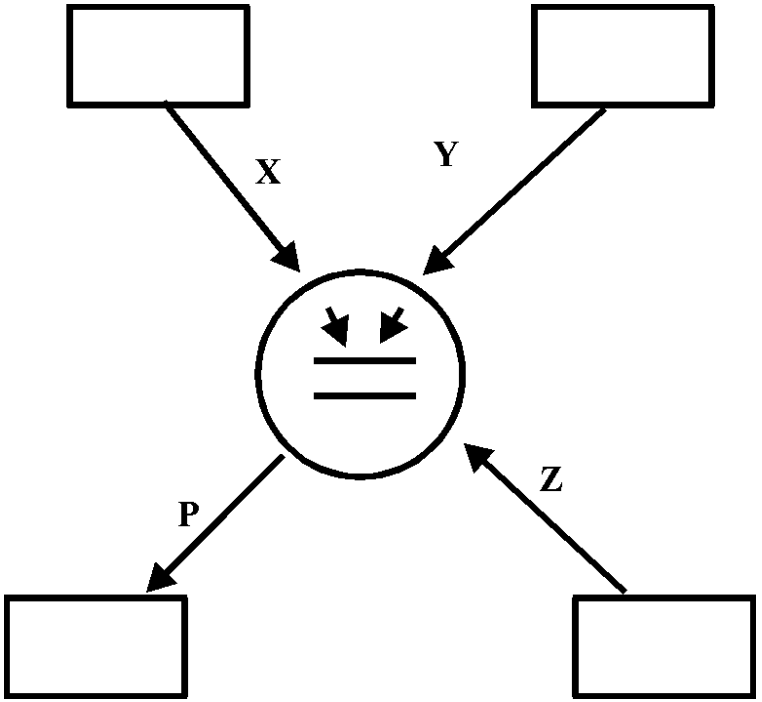


2) pro řešení události požadovány
další údaje z okolí systému (Y,Z)

Události a průvodní toky

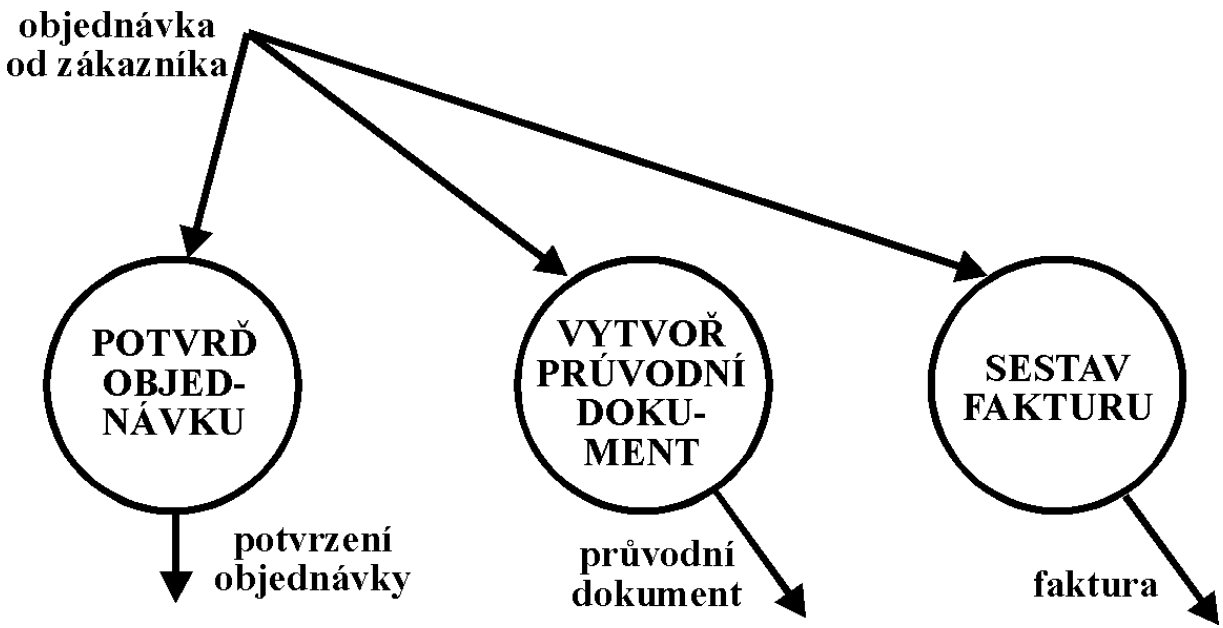


3) odezva na událost ve formě výstupního toku (P)



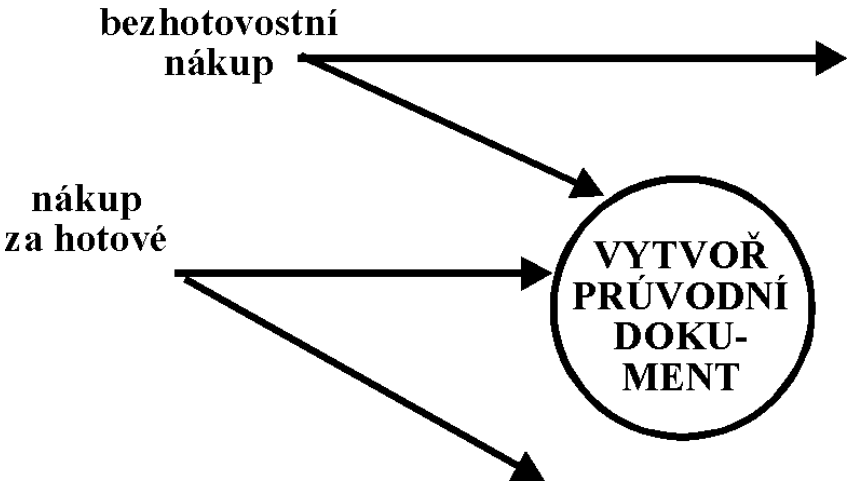
4) odezva na událost ve formě uložených dat

Události s více odezvami



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

Události s více odezvami



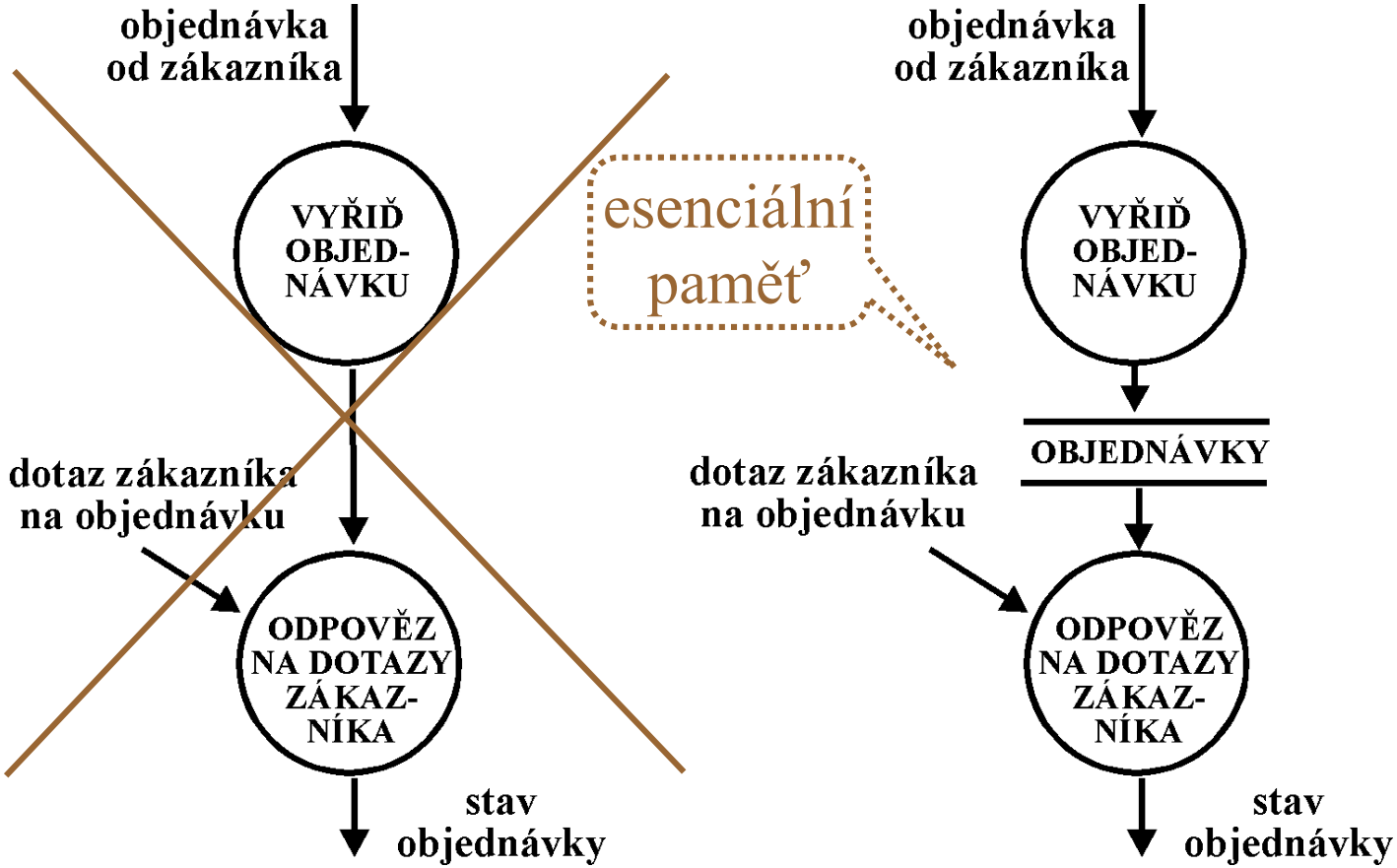
Odezva produkovaná procesem musí být identická pro různé události.
Vstupní a výstupní data jsou identická pro různé události.

Propojení odezev na události

Odezva na událost může vyžadovat data vytvořená jinou událostí.
Neznáme, ve kterém časovém okamžiku události nastanou.
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.

Časově zpožděná komunikace



Prvotní model chování

prvotní DFD ⟺ prvotní ERD

paměti ⇄ entity

Kontrola:
- Zpracovává prvotní DFD každou událost?
- Jsou zakresleny všechny potřebné vstupy a výstupy pro každou událost?
- Jsou zakreslena nezbytná propojení mezi událostmi?

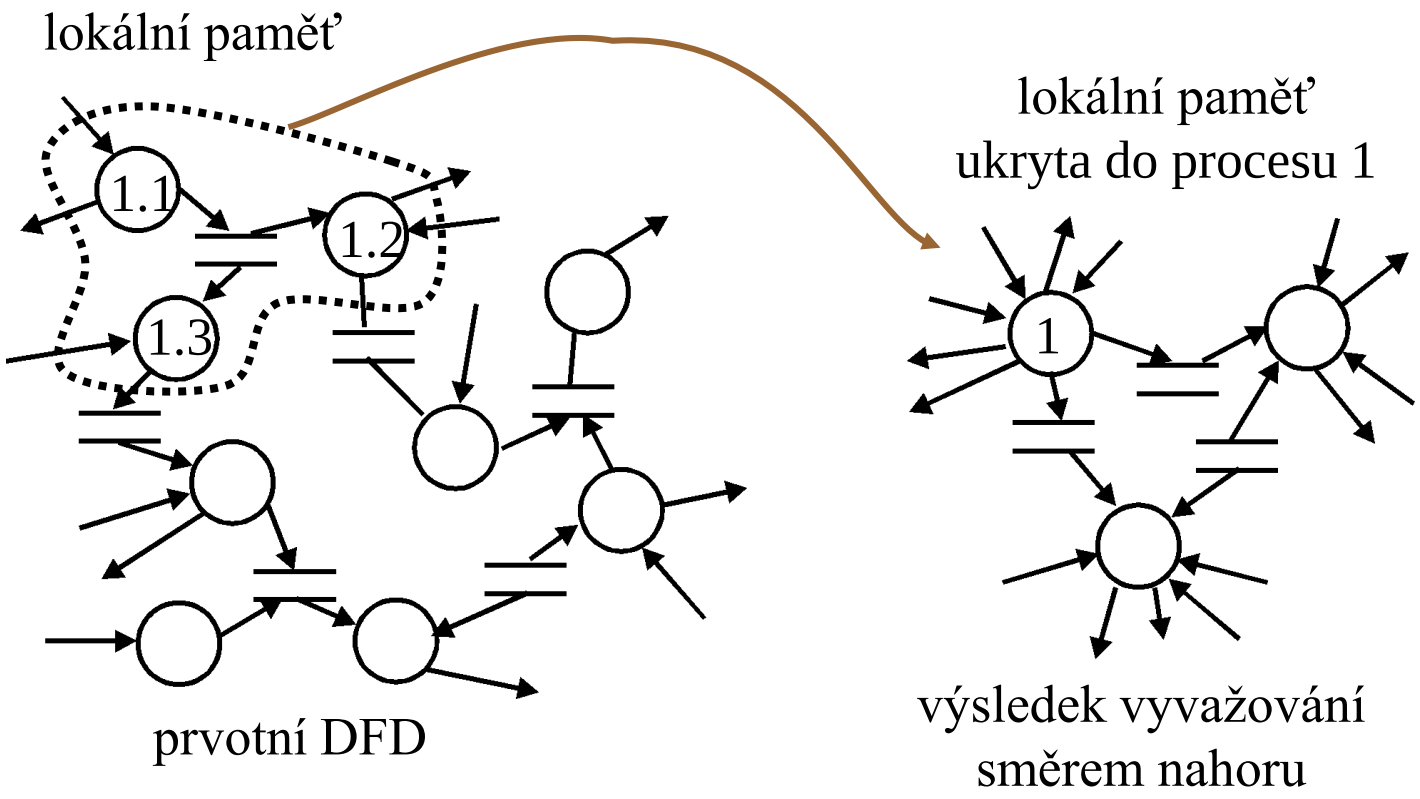
Prvotní model chování nekonzultujeme s uživatelem !
Model není uspořádaný tak, aby mohl být pochopen jako celek.

Vyvažování směrem nahoru

Hledáme seskupení vzájemně souvisejících procesů
do agregovaného procesu na diagramu vyšší úrovně.

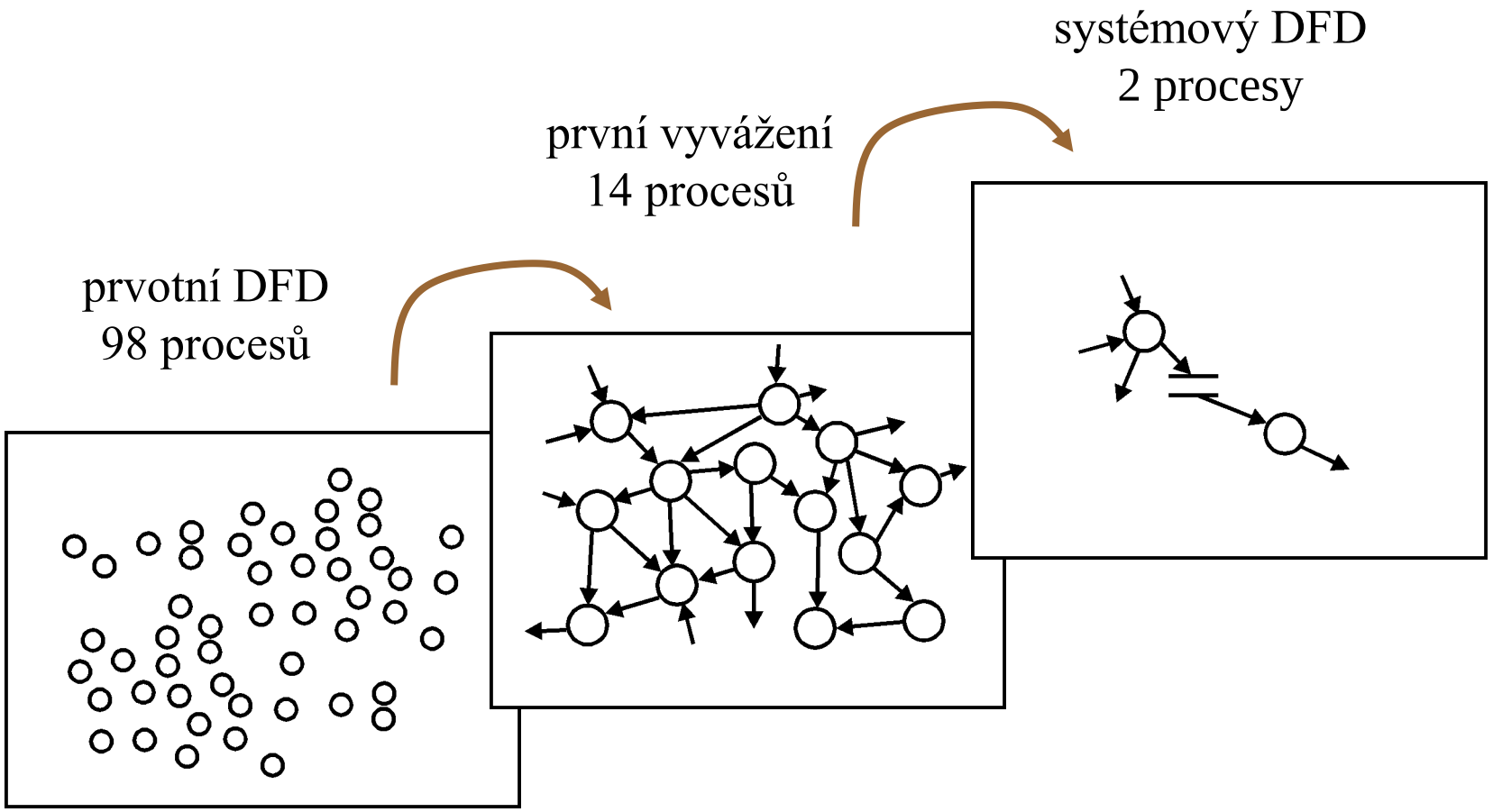
1. Seskupení procesů má zahrnout blízké, obdobné odpovědi (obdobně pojmenované procesy).
2. Paměti zakrýváme, pokud paměť používá pouze skupina procesů na nižší úrovni (žádný jiný proces vně této skupiny paměť nepoužívá).

Vyvažování směrem nahoru



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

Vícenásobné vyvažování směrem nahoru

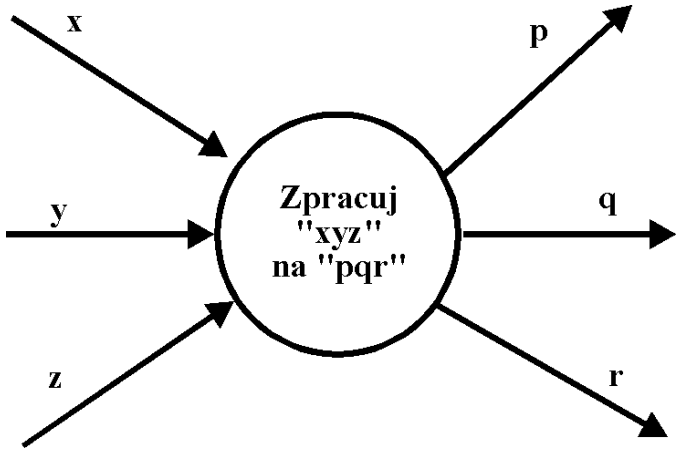


Vyvažování směrem dolů

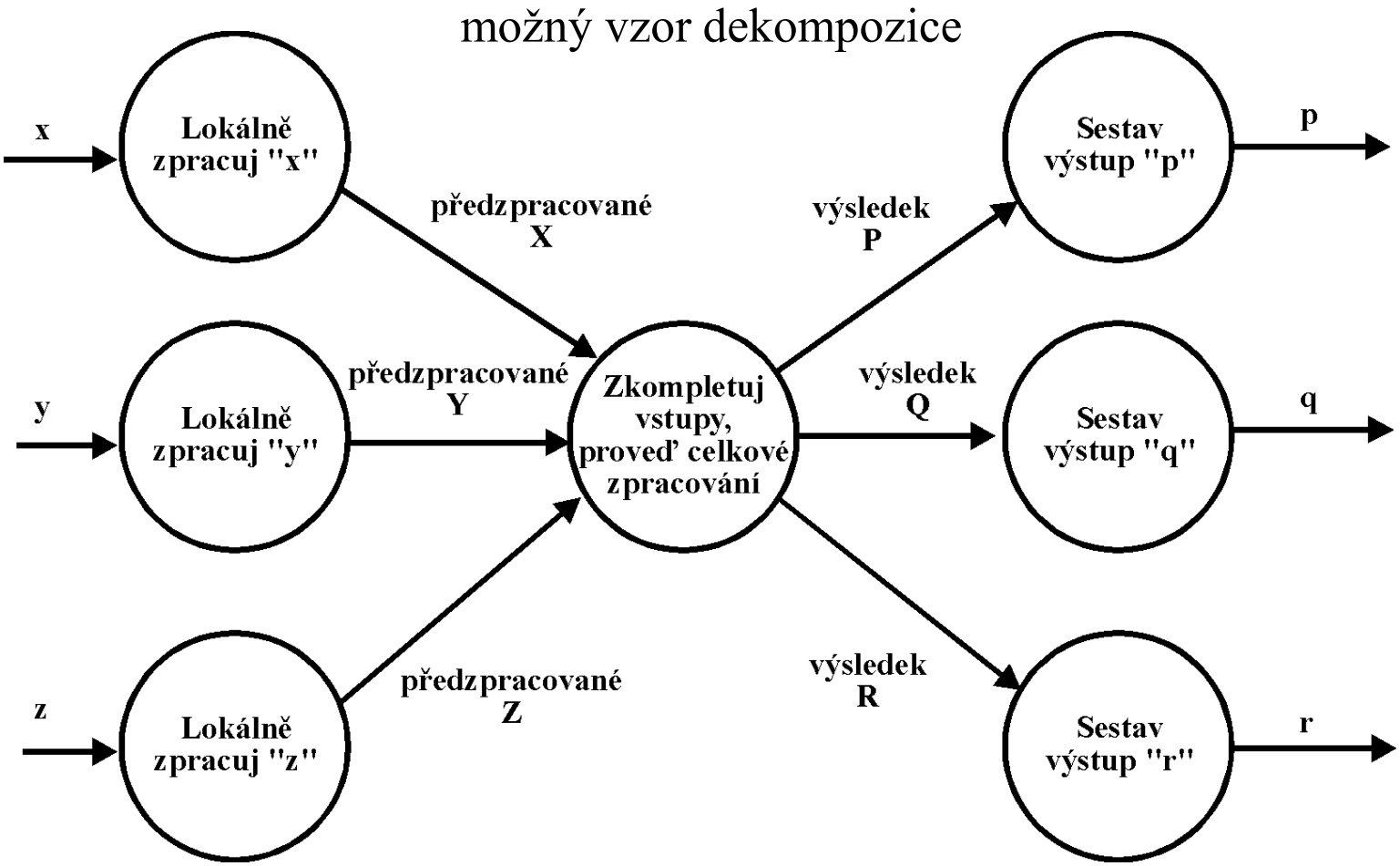
Čistá funkční dekompozice: u procesu se složitou funkcí jsou zřetelné dílčí funkce. Tyto funkce vyjádříme jako procesy na nižší úrovni DFD.

Funkční dekompozice nezřetelná: Pokusíme se odvodit dekompozici na základě vstupních a výstupních toků.

Data určují směr dekompozice.



Vyvažování směrem dolů



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