

Softwarové inženýrství
Strukturovaná analýza
Předmět Informační systémy

Daniela Szturcová

Softwarové inženýrství

- Princip modelování – podpora grafickými prvky
- Princip iterace – dnes prototypový přístup
- Princip strukturování – rozklad na části
- Princip životního cyklu – celý proces, nástroje
- Princip automatizace – podpora počítačovými technikami během celého cyklu

Informační inženýrství

- 80. léta 20. stol. J. Martin
- souvislost s výstavbou velkých databázových systémů – tvorba informačního modelu podniku
- Hypotézy:
 - informace = “srdce podniku”
 - data = nejstabilnější složka
- hlavní role – administrátor, správce dat

Organizační inženýrství

Systematický inženýrský přístup k budování prosperující a výkonné organizace podniku založené na pružné a objektivní práci s aktuálními informacemi.

Spojení metod informačního inženýrství a organizačních metod a praktik – základ:

KISS – Komunikace, Informace, Systémovost a Standardizace

Budování organizace ve dvou krocích:

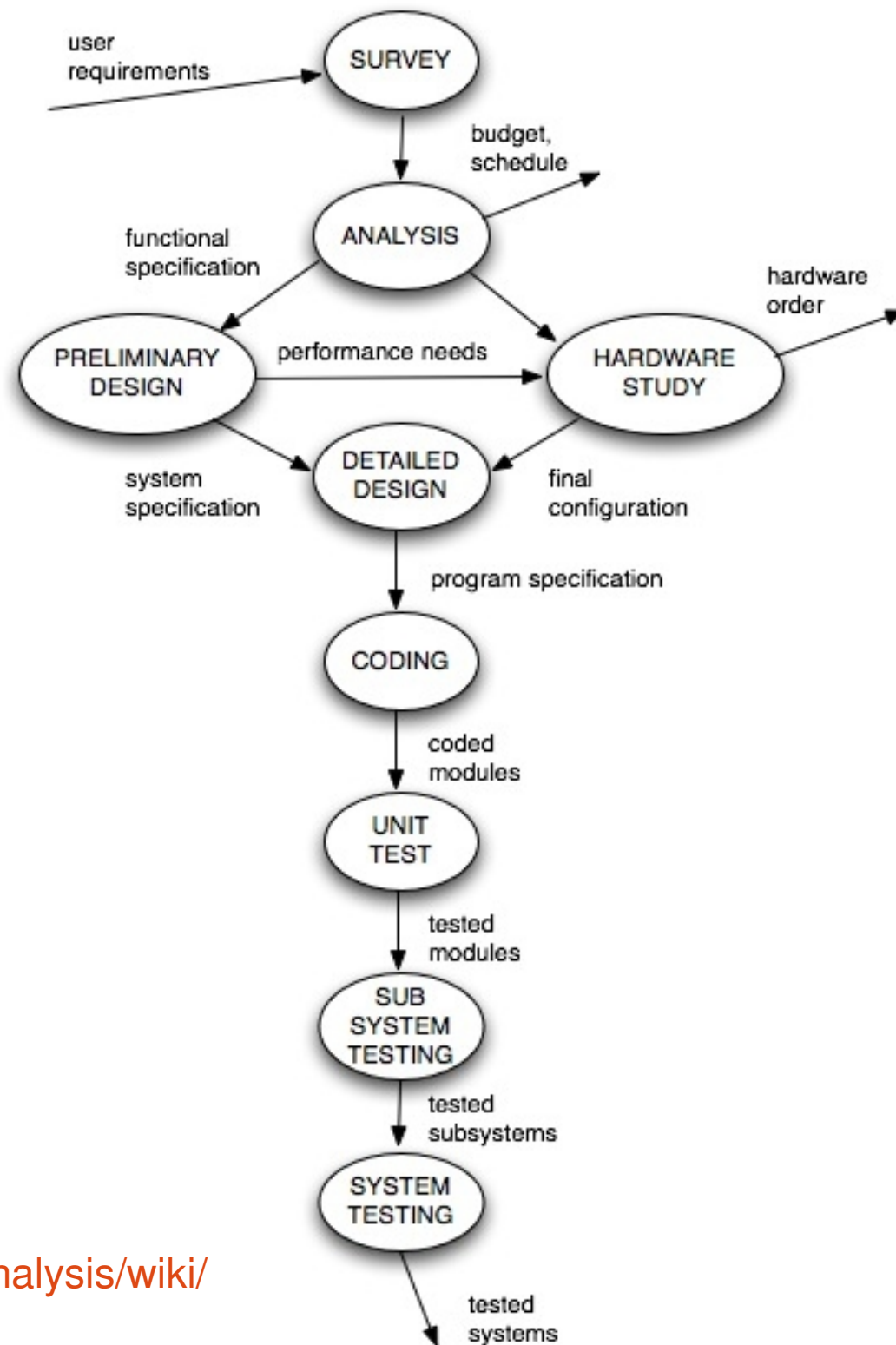
- diagnostika stavu a identifikace problémů
- zlepšování stavu, jeho kultivace

Organizační inženýrství - Organization Performance model

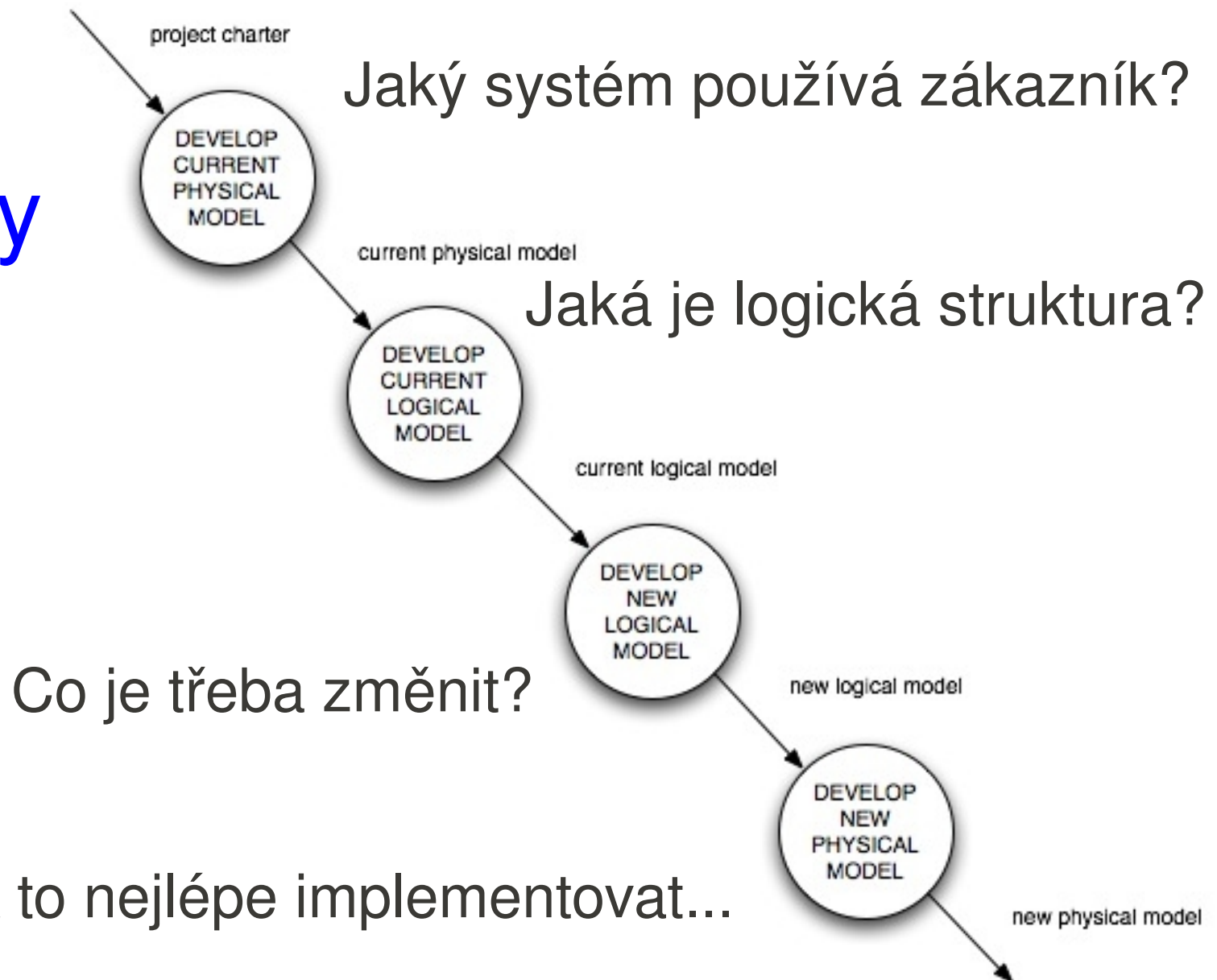
Systematická kultivace firmního prostředí

- Zhodnocení situace v organizaci vzhledem k jejímu okolí
- Stanovení nebo korekce strategických cílů
- Formování:
 - identifikace činností
 - přiřazení lidí k činnostem
 - vytvoření struktury
 - návrh principů oceňování důležitosti činností
 - identifikace informačních potřeb
 - rozdělení kompetencí
- Budování kultury organizace
- Analýza výsledků působení organizace

Klasický vývoj projektu



Vývoj projektu 4 modely



DeMarcova metodologie

Vstup: uživatelské požadavky

Výstup: tzv. strukturovaná specifikace

- Systém je specifikován pomocí DFD, jsou uvedeny podstatné procesy, paměti a údaje.
- Jednodušší procesy v DFD nižší úrovně, elementární procesy zapsány v pseudokódu, rozhodovací tabulkou nebo stromem.
- V datovém slovníku je uložen popis dat.

DeMarcova metodologie

Fyzický model stávajícího systému

- slouží k poznání současného stavu systému,
- analytik zmapuje stávající systém, jeho funkční strukturu a data, která systém obsahuje,
- model může obsahovat zpracování a přesun i fyzických formulářů apod.

DeMarcova metodologie

Logický model stávajícího systému

- z fyzického modelu vytvoříme logický (cca 75% redukce),
- zrušíme všechny implementační detaily, modelujeme co by systém dělal, kdybychom měli ideální technologii (nekonečné paměti, nekonečnou rychlost atd.),
- získáme logické procesy a podstatu transformace dat,
- model obsahuje i funkce a data, která nevyhovují.

DeMarcova metodologie

Logický model nového systému

- většina systému pravděpodobně zůstane stejná,
- zpracují se požadavky na nové funkce,
- po konzultaci s uživatelem promítnuty změny do logického modelu.

Fyzický model nového systému

- návrh implementace.

Všechny 4 modely vyjadřují uživatelský pohled.

DeMarcova metodologie

Při tvorbě fyzického modelu - uživatel ví o systému víc než analytik; často vznikne dojem, že analytik problematice nerozumí.

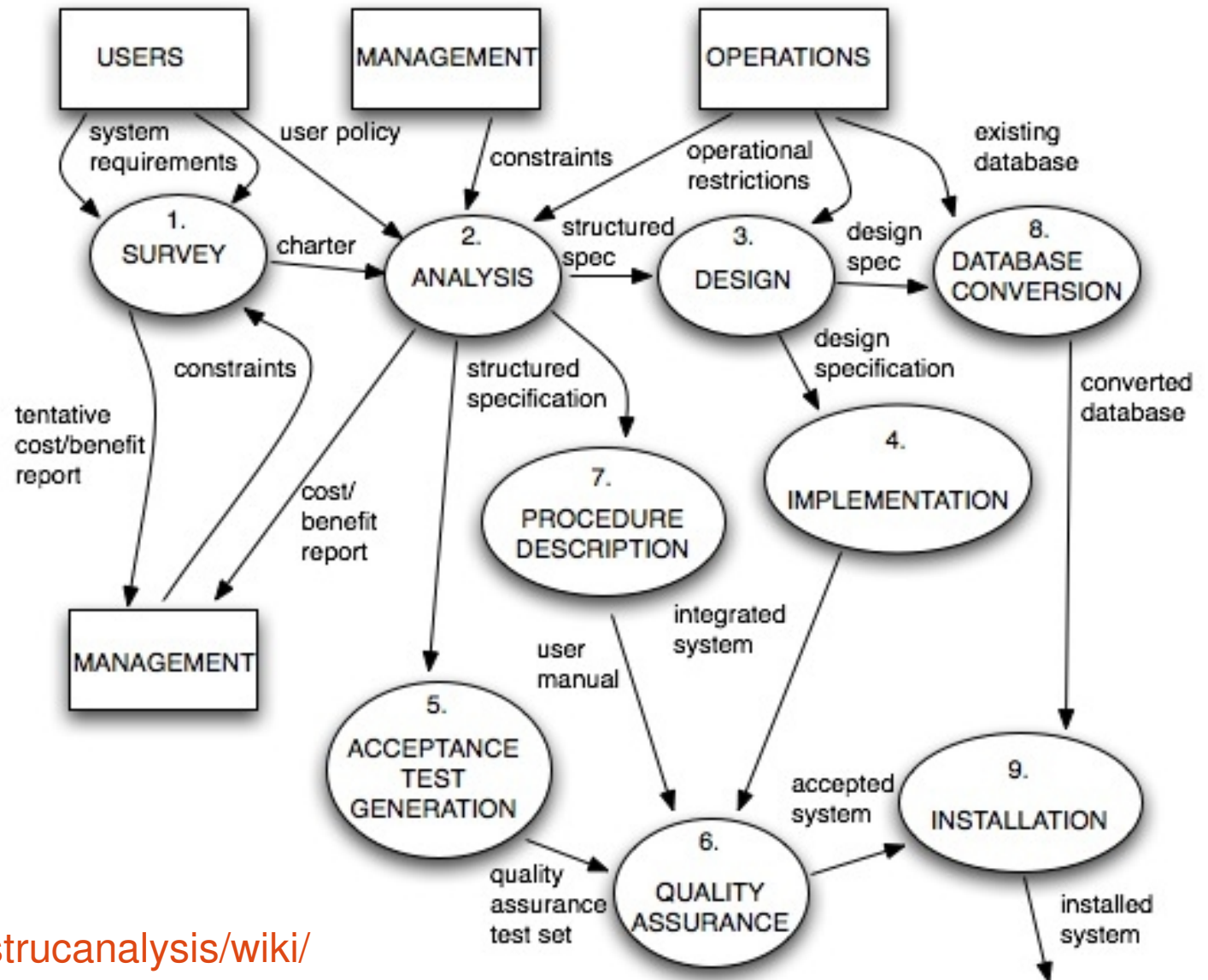
Uživatel odmítá spolupráci na vývoji nového logického modelu:

- analytik neumí sám vytvořit fyzický model stávajícího systému,
- nemůže pak ani dobře navrhnout nový systém.

Tvorba 4 modelů snižuje ochotu spolupracovat s analytikem.
(4 modely je prostě moc).

Problémy se snaží řešit "moderní strukturovaná analýza"
(Yourdon).

Strukturovaný vývoj projektu



Yourdon

Ed Yourdon v roce 1989 navrhl místo čtyř modelů systému jiný postup. Najít esenci systému, tj. logický model nového systému.

Pomocí osvědčených technik a přístupů vyvinutých na konci 70. a v 80. letech navrhl postup tvorby tzv. *esenciálního modelu*.

Esenciální model je založen na myšlence - **CO** má systém dělat, aby splnil požadavky a potřeby uživatelů.

Yourdonův esenciální model

Model prostředí (environmental model):

- účel systému
- kontextový DFD
- seznam vnějších událostí

Model chování systému (behavioural model):

- víceúrovňový DFD, minispecifikace
- ER model
- DD (datový slovník)
- STD (State Transition Diagram)

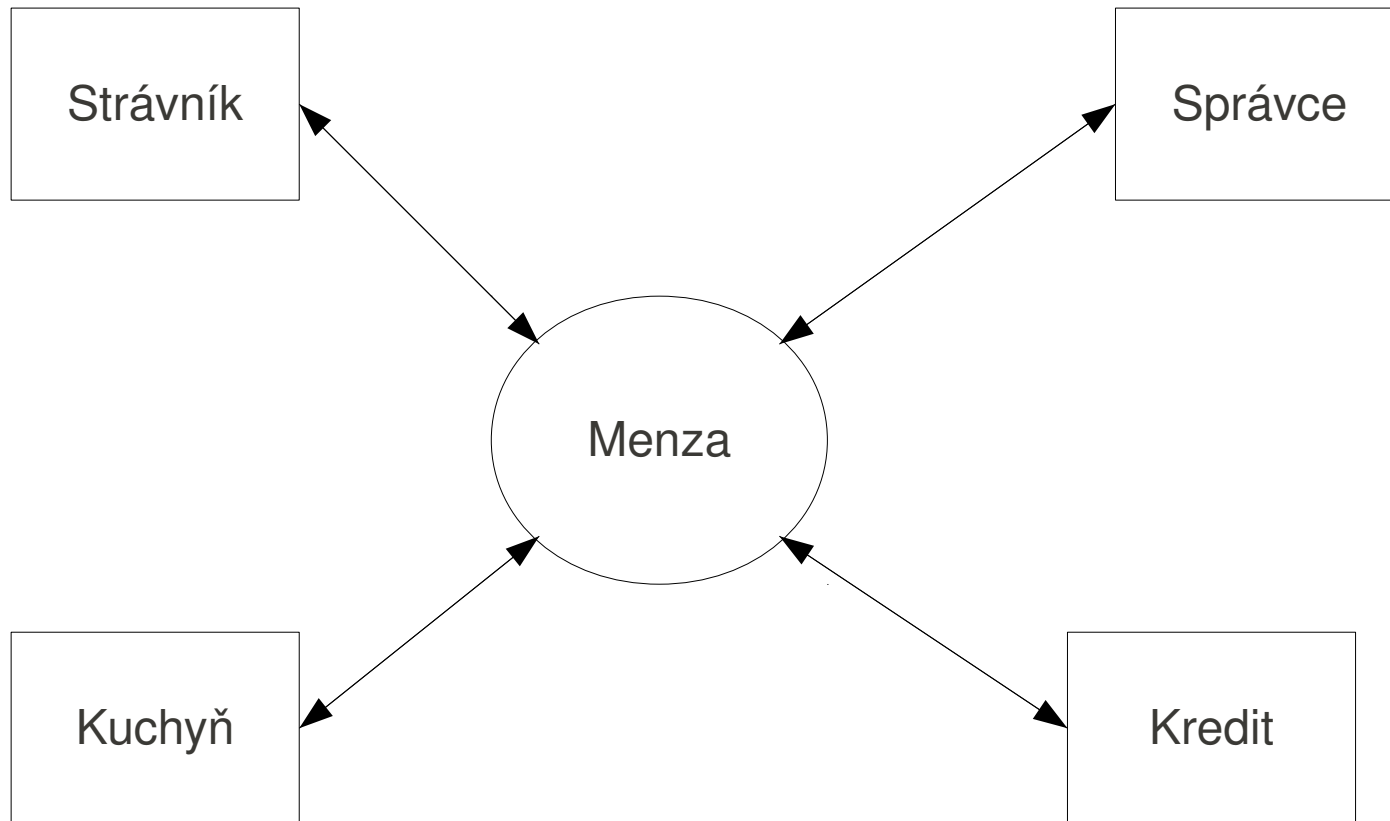
Postup tvorby modelu

Model prostředí

(definuje hranici mezi systémem a zbytkem světa)

- definujeme účel systému (ne delší než odstavec)
- vytvoříme kontextový model systému
- vytvoříme seznam událostí

Kontextový diagram



Postup tvorby modelu

Seznam událostí - identifikují se všechny externí události, na které systém musí reagovat.

Vytvoří se první předběžný DFD.

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 zaměstanců	Datový	Založí/změní/zruší účet, obnoví stav zaměstanců

Tvorba seznamu událostí

Klasifikace událostí:

- **událost datového toku** (Flow, Datový) – událost, která se projeví příchodem dat.

Například: Událost "přišla objednávka"

- **časová událost** (Temporal, Časový)

Například: Zpracování transakcí mezi bankovními účty nastane v 3:00.

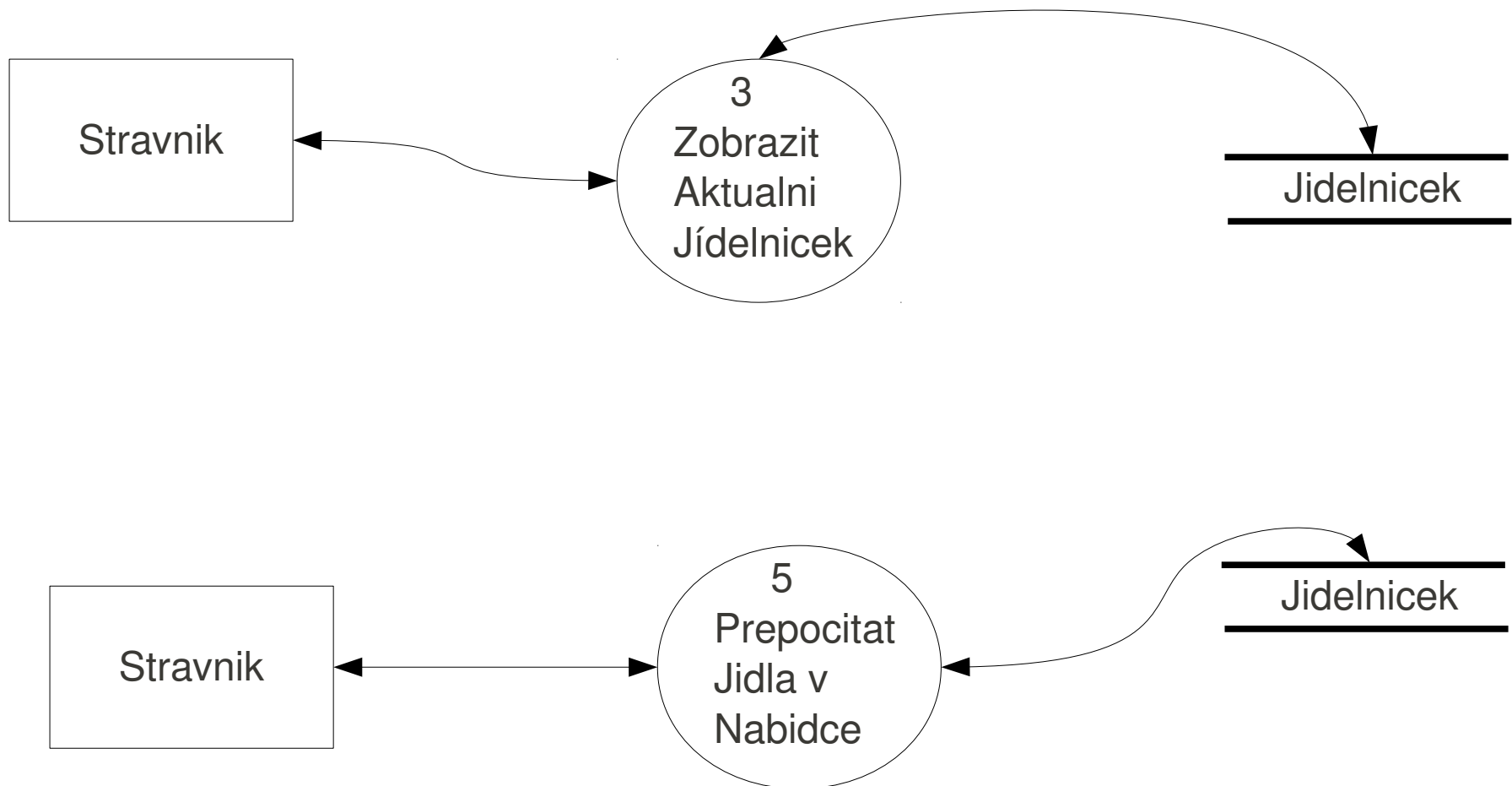
- **řídící událost** (Control) - asynchronní událost, například v real time systémech.

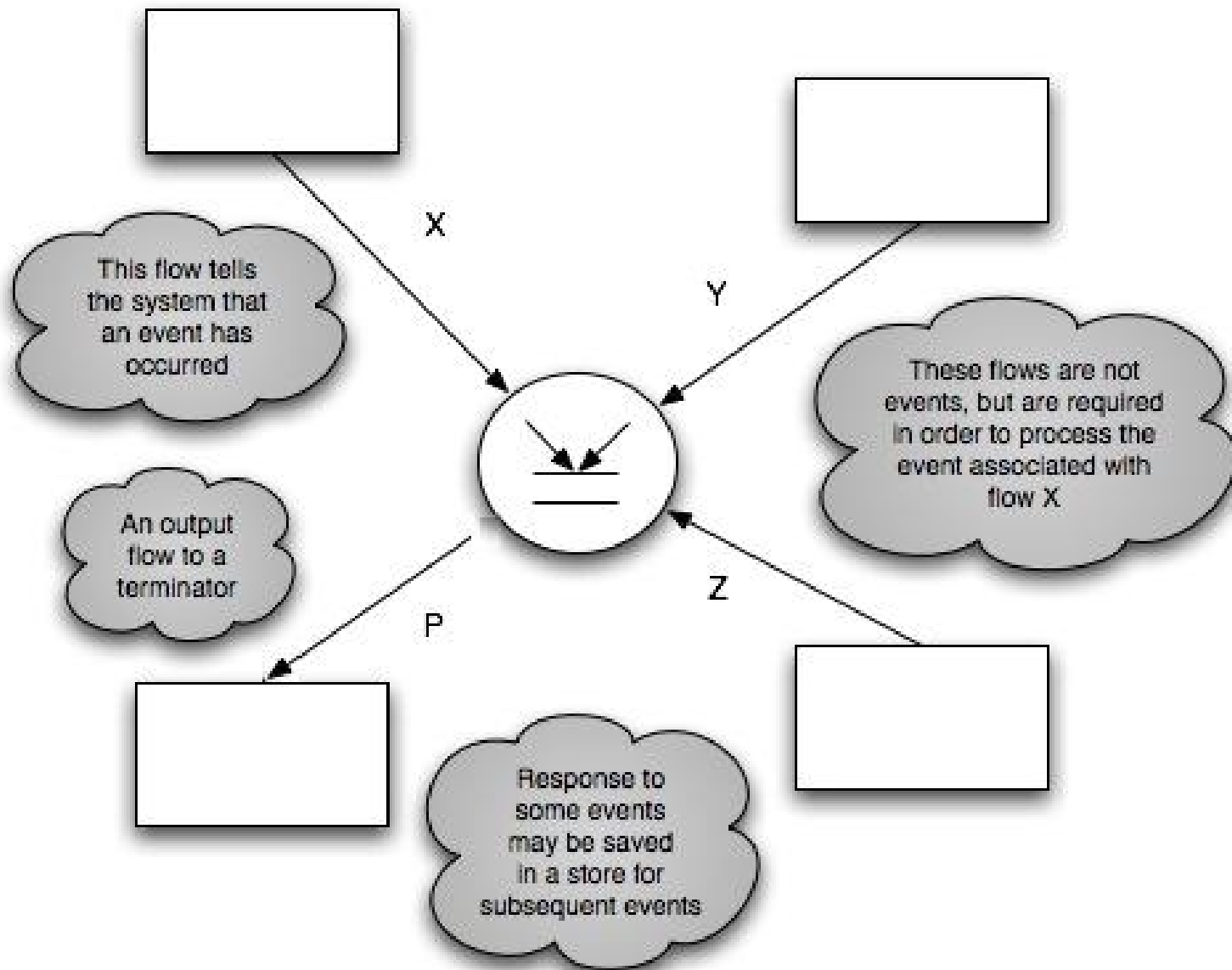
Postup tvorby modelu

Pro každou událost se nakreslí jeden DFD:

- pro každou událost ze seznamu nakreslíme bublinu, očíslováme jí podle čísla události
- bublinu pojmenujeme podle odpovědi na událost (např. odpověď na "zákazník zaplatil fakturu" je "vložit platbu mezi příjmy")
- k bublině nakreslíme vstupy, výstupy a potřebné paměti
- identické bubliny sdružíme do jedné (identické bubliny mají stejný vstup, výstup a proces)
- výsledný DFD zkontrolujeme oproti kontextovému diagramu a seznamu událostí

DFD k události





Postup tvorby modelu

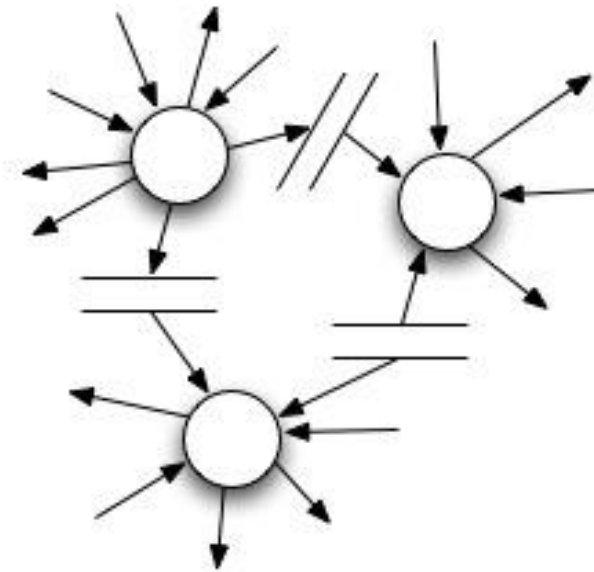
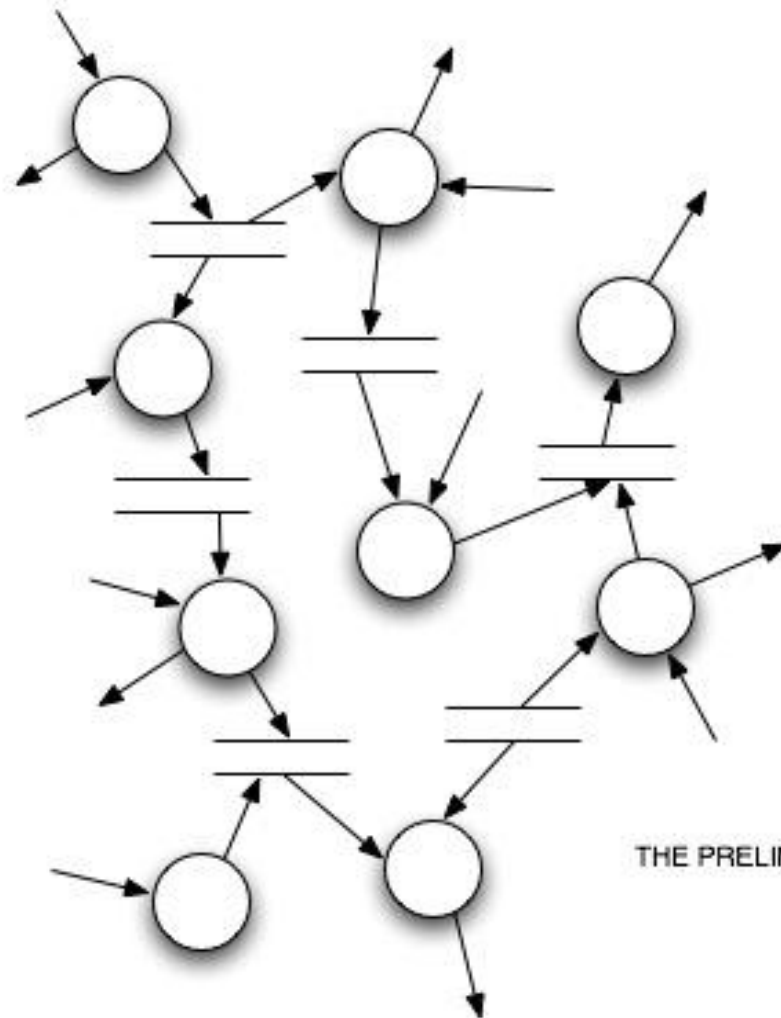
Vytvořený předběžný model transformujeme na víceúrovňový, pomocí tzv. **level balancing**

- směrem nahoru (ukrývání informace)
- nebo směrem dolů (rozdělení na podrobnější informace)

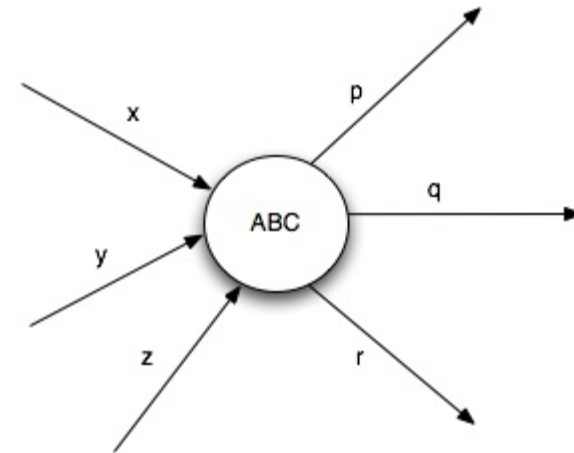
Level balancing

- strukturování - agregujeme spřízněné procesy, sdružíme je do bublin ve vyšší úrovni DFD (skupiny po cca 7+/- 2 procesech a pamětech),
- každý agregát se týká příbuzných odpovědí na události, většinou obsahuje procesy zpracovávající příbuzná data
- pokud procesy nejsou primitivní a vyžadují další rozdělení (většinou jsme bublinu nedokázali dobře pojmenovat), pak je rozdělíme na primitivní procesy

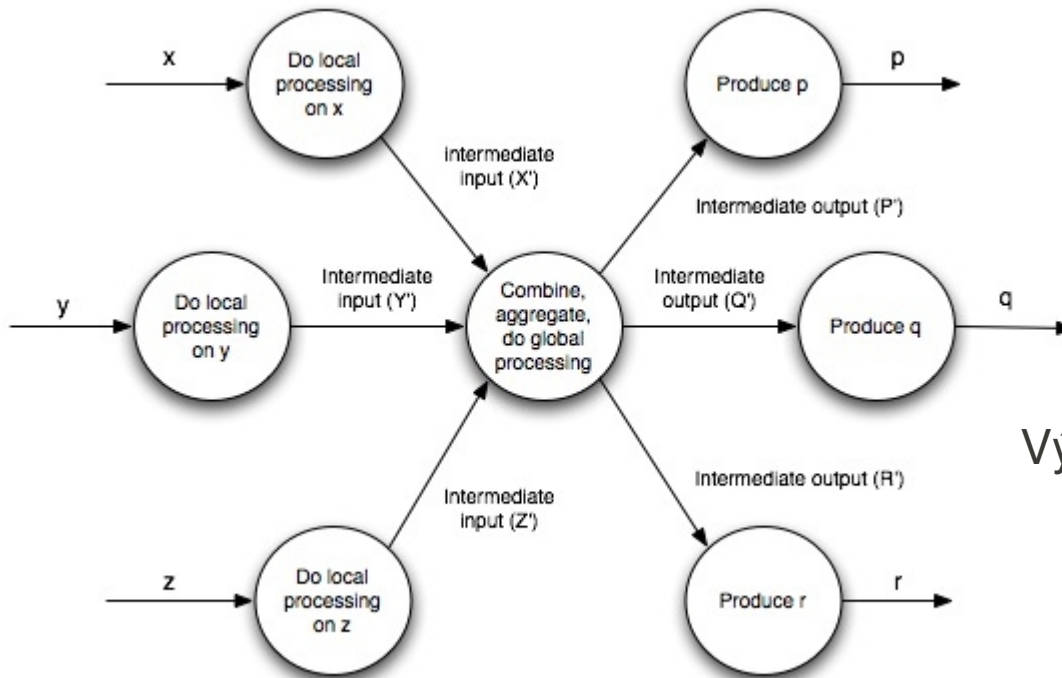
Level balancing – směrem nahoru



Level balancing – směrem dolů



Počáteční DFD



Výsledek balancování směrem dolů

Dokončení esenciálního modelu

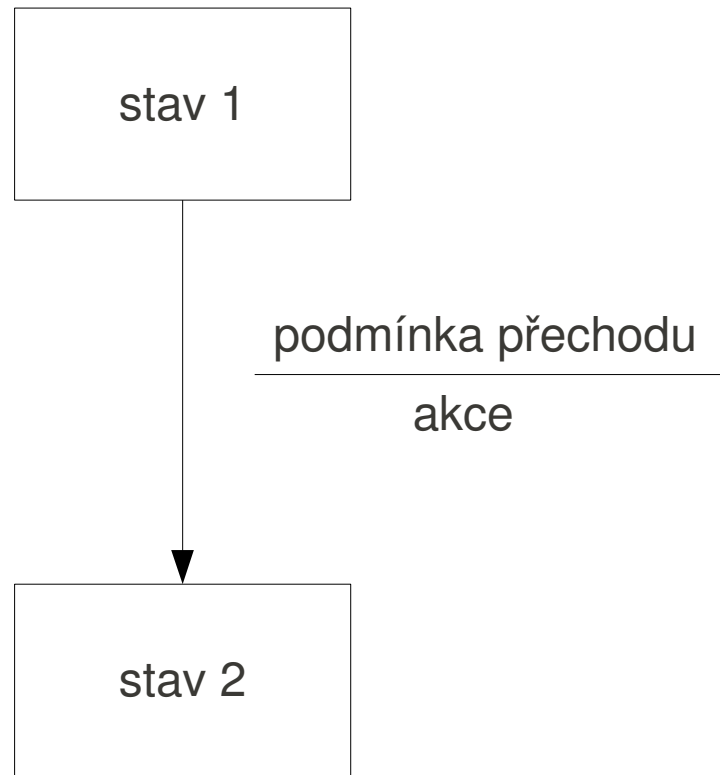
- Vytvoříme (mini)specifikaci procesů.
- Dokončíme datový slovník, ER model.
- Vytvoříme STD

Uvedené informace tvoří tzv. esenciální model systému.

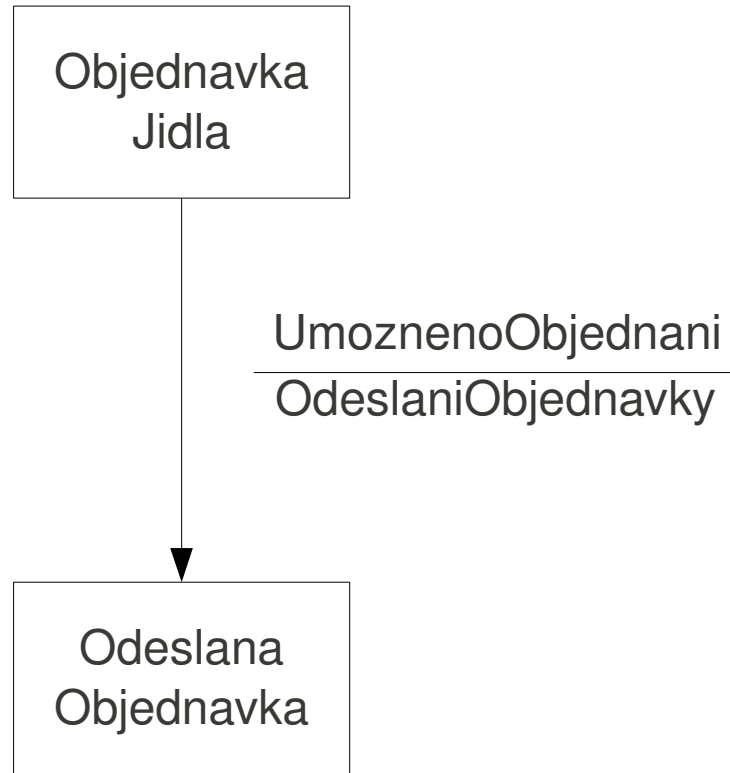
State Transition Diagram (STD)

- **Stav systému** charakterizuje vlastnosti systému v daném okamžiku, graficky obdélník.
- Změna stavu představuje **přechod systému** z jednoho rozpoznatelného stavu do druhého. Graficky šipka s ohodnocením, které popisuje **podmínku změny stavu** a **akci** spojenou s touto změnou.

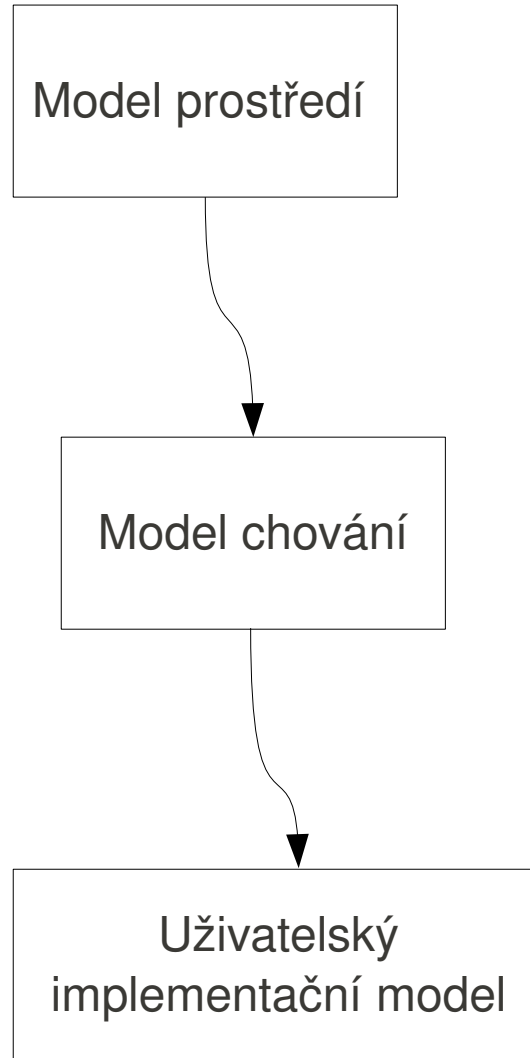
STD



STD



Postup tvorby modelu



Uživatelský model systému

- Rozdělení esenciálního modelu mezi lidi a počítač (které procesy a paměti budou realizovány manuálně) – rozhoduje uživatel.
- Rozhraní aplikace s uživatelem (volba vstupních a výstupních zařízení, formáty obrazovek, formáty výstupů).
- Mimofunkční požadavky na aplikaci - omezení, např. objemy dat, časy odpovědí na různé události atd.

Implementační model systému

Posledním krokem YMSA je vytvoření implementačního modelu. Přizpůsobuje esenciální model konkrétnímu nasazení.

- Výhoda – při změně technologie se změní pouze implementační model, esenciální zůstane stejný.

Při analýze nemusí být nutně použity všechny nástroje (modely) - používají se jen ty, které mají v daném kontextu smysl.

Zdroje

- Yourdon, Ed: Modern Structured Analysis (Prentice Hall, 1989)
- <http://yourdon.com/strucanalysis/wiki/>
- Jilková, H., Stanovská, I.: Strukturované postupy analýzy a návrhu informačních systémů