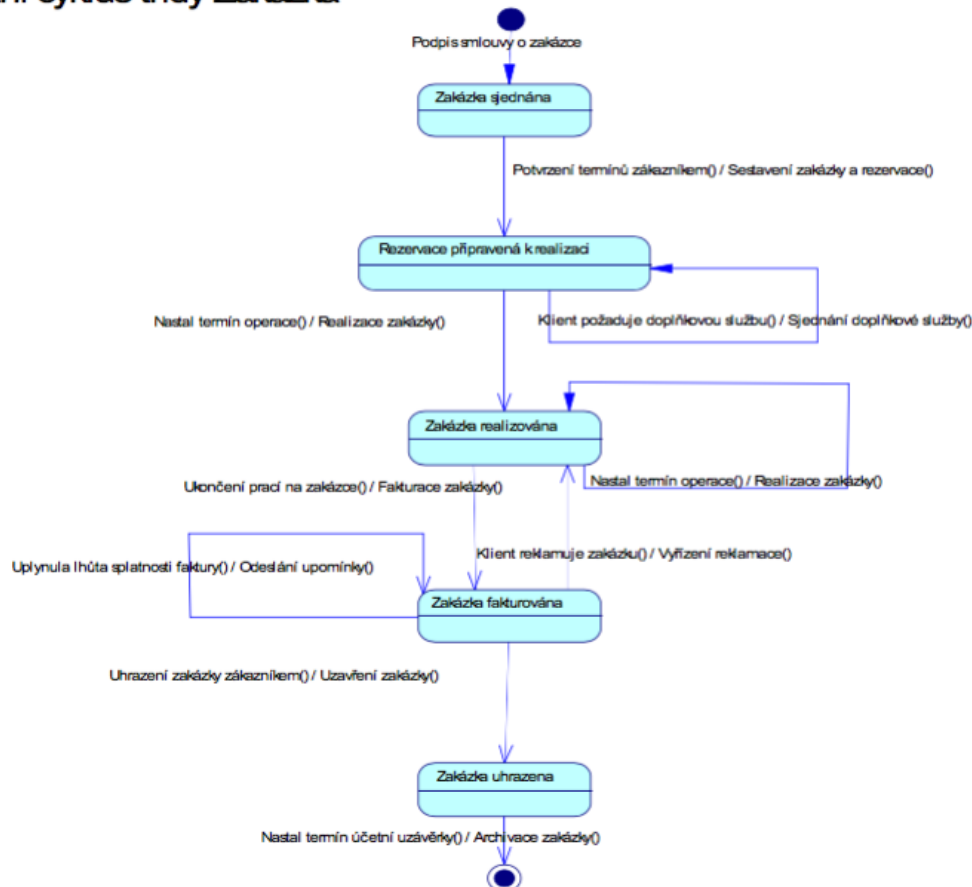


Poznámky k IMO

- Klíčový proces přináší společnosti hodnotu
- Životní cyklus objektu popisuje všechno možné chování objektu - všechny možné posloupnosti volání metod
- Volání metod jinak, než popisuje statechart je nepřipustná operace
- Každá nepřimitivní třída má právě jeden statechart
- Každá metoda třídy je použita u přechodu v statechartu a naopak každý přechod v statechartu je popsán metodou třídy
- <<konstruktor>> je použit u přechodu z počátečního stavu a <<destruktory>> u přechodů do koncových stavů, ostatní přechody používají <<transformery>>

Příklad Statechartu

Životní cyklus třídy ZAKÁZKA



Konzistence

Provázání procesů s objekty

- Každá třída objektů z modelu tříd musí být zastoupena v modelu procesů v alespoň jednom z jeho vstupů, či výstupů a/nebo aktérů, či jiných externích aspektů.
- Každý vstup, či výstup procesu, jakož i každý externí aspekt procesu, musí být zastoupen v modelu tříd jako třída, nebo asociace mezi třídami, či jako

kombinace obojího.

- Každá událost, specifikovaná v popisech přechodů ve stavovém diagramu životního cyklu třídy, musí korespondovat s událostí, specifikovanou v popisu nějakého (nějakých) business procesu (procesů).

DFD

- Názvy procesů - dostatečně obecné - srozumitelné uživateli
- Očíslovat procesy - číslo identifikuje proces v rámci úrovně
- Číslo určuje příslušnost procesu do nadřazeného procesu
- Jeden DFD 7+-2 procesy - vždy však musí být úplný
- DFD nesmí obsahovat černé díry a samogenerující procesy, neoznačené toky a procesy, read only, write only data story

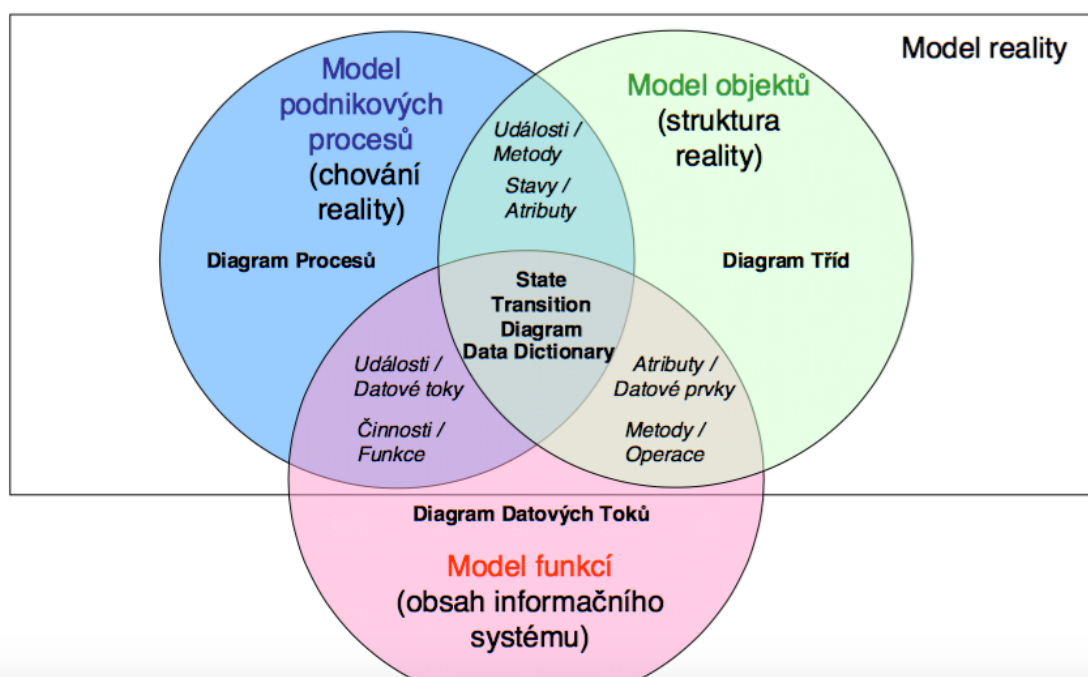
Event partitioning approach

- Pro každou událost vytvořit proces
- Každý proces pojmenovat podle reakce systému na událost
- Ke každému procesu doplnit vstupy a výstupy, případně data story
- Kontrola konzistence
- Rozpustit do subdiagramů

Pravidla konsistence DFD (konsistenční omezení vztahů metatříd)

- DataStore musí mít alespoň jeden vstupní DataFlow a jeden výstupní DataFlow.
- DataFlow smí spojovat pouze Funkci a Funkci, Funkci a DataStore nebo Terminátor a Funkci.
- DataFlow Terminátor -> Funkce musí mít přiřazenu událost
- Funkce musí mít alespoň jeden DataFlow

IS jako model reality



DFD

Cílem je popsat funkčnost informačního systému

- funkčnost = potenciál chování systému
- cílem chování informačního systému je **odrážet dění v reálném (business) systému – viz Princip modelování, neboli modelovat relevantní kombinace událostí a akcí** ať obecně platné (neb nutno respektovat obecná „business rules“) tak záměrné (neb nutno podporovat „business procesy“)

Provázání DFD s objekty

- Každý elementární Datastore v DFD musí být v CD zastoupen jako třída, nebo asociace, nebo kombinace obojího
- Atributy každého elementárního datastore z DFD musí být datovou strukturou atributů tříd, jimiž je tento Datastore v CD zastoupen
- Metody každé elementární funkce z DFD musí být algoritmickou strukturou metod tříd, jimiž jsou v CD zastoupeny Datastorey, spojené datovými toky s touto funkcí

Provázání DFD s procesy

- Každý proces má vazbu alespoň na 1 funkci
- Každá funkce má vazbu alespoň na 1 proces
- Každá událost v procesním modelu má vazbu na vstupní tok v DFD
- Každý elementární vstupní tok v DFD od terminátoru (tj. zevnějšku systému) musí odpovídat nějaké události, specifikované v popisu nějakého (nějakých) business procesu (procesů) v PD
- Každý stav každého procesu v PD musí korespondovat s některými elementárními Datastorey v DFD a naopak každý elementární Datastore v DFD musí korespondovat s některými stavy procesů v PD. Jde o korespondenci M:N

Další poznámky

- Životní cyklus objektu je pohled na reálný systém jako na strukturu objektů a jejich podstatných (stálých, relativně neměnných) vztahů – tento pohled představuje model tříd (business) objektů, pohledem na reálný systém jako na strukturu vzájemně navazujících činností, zpracovávajících vstupy (suroviny) na výstupy (produkty), a to v zájmu definovaného cíle – tento pohled představuje model (business) procesů.
- Funkce v DFD má jednoznačný identifikátor - hierarchický - např. 5.1.2 - nadřazené funkce
- Jasně popisy dat, která tečou v DFD
- Při udržování konzistence dat mezi DFD a CD je nutné především udržovat vazbu mezi každým Data Store a nějakou danou strukturou tříd a jejich vazeb v CD. Dále je vhodné udržovat nějakou vazbu mezi metodami třídy a toky dat v DFD. Tato konzistentní vazba je však již mnohem méně těsná a proto je možné ji při technických obtížích s jejím zaznamenáváním do modelovacího nástroje oželeť.

- A) Každá třída objektů z modelu tříd musí být zastoupena v modelu procesů v alespoň jednom z jeho vstupů, či výstupů a/nebo aktérů, či jiných externích aspektů.
- B) Každý vstup, či výstup procesu, jakož i každý externí aspekt procesu, musí být zastoupen v modelu tříd jako třída, nebo asociace mezi třídami, či jako kombinace obojího.
- C) Každá třída objektů z modelu tříd musí mít specifikovány alespoň tři metody:
- metodu, jíž objekt (instance třídy) vzniká (konstruktor),
 - metodu, jíž objekt (instance třídy) zaniká (destruktor),
 - alespoň jednu metodu, jíž se mění atributy objektu (transformer).
- D) Pro každý atribut třídy objektů z modelu tříd musí být u této třídy specifikována metoda, jíž je tomuto atributu přidělena počáteční hodnota a metoda, jíž je hodnota atributu změněna.
- E) Pro každou asociaci mezi třídami musí být u každé takto asociované třídy specifikována metoda, odpovídající této asociaci.
- F) Ke každé třídě objektů, která není považována za primitivní, je přiřazen stavový diagram, popisující její životní cyklus.
- G) Stavový diagram životního cyklu třídy musí mít popsány všechny přechody mezi všemi svými stavy. Popis každého přechodu specifikuje jednak událost, na základě které se přechod uskuteční (spoušť), jednak metodu, jíž se tento přechod uskuteční.
- H) Stavový diagram životního cyklu třídy musí obsahovat v popisech přechodů mezi stavy všechny metody této třídy. Popisy přechodů mezi stavy nesmí obsahovat metody, které nejsou specifikovány u této třídy.
- I) Každá událost, specifikovaná v popisech přechodů ve stavovém diagramu životního cyklu třídy, musí korespondovat s událostí, specifikovanou v popisu nějakého (nějakých) business procesu (procesů).
- J) Každý elementární Datastore¹⁷ v DFD musí být v CD zastoupen jako třída, nebo asociace, anebo kombinace obojího.
- K) Atributy každého elementárního Datastore z DFD musí být datovou strukturou atributů tříd, jimiž je tento Datastore v CD zastoupen.
- L) Metody každé elementární funkce¹⁸ z DFD musí být algoritmickou strukturou metod tříd, jimiž jsou v CD zastoupeny Datastorey, spojené datovými toky s touto funkcí.
- M) Každý elementární vstupní datový tok v DFD od terminátoru (tj. zvnějšku systému) musí odpovídat nějaké události, specifikované v popisu nějakého (nějakých) business procesu (procesů) v PD.
- N) Každý stav každého procesu v PD musí korespondovat s některým(i) elementárním(i) Datastorem(y) v DFD a naopak každý elementární Datastore v DFD musí korespondovat s některým(i) stav(y) procesů(ů) v PD. Jde o korespondenci M:N.

Specializace diagramu tříd na DFD je realizována zavedením 4 standardních stereotypů:

- DataStore,
- Funkce,
- Terminátor,
- DataFlow.

První tři jsou stereotypy objektu třída, DataFlow je stereotypem asociace.

Pro takto stereotypizovaný model tříd pak platí pravidla konzistenčních vztahů v DFD:

- DataStore musí mít alespoň jeden vstupní DataFlow a jeden výstupní DataFlow.
- DataFlow smí spojovat pouze Funkci a Funkci, Funkci a DataStore nebo Terminátor a Funkci.
- DataFlow Terminátor -> Funkce musí mít přiřazenu událost
- Funkce musí mít alespoň jeden DataFlow

C.2.10.4 Ostatní doplňková pravidla

- R) Každý proces má vazbu alespoň na 1 požadavek
- S) Každý požadavek má vazbu alespoň na 1 proces
- T) Každá funkce má vazbu alespoň na 1 požadavek
- U) Každý požadavek má vazbu alespoň na 1 funkci
- V) Každá událost v STD by měla mít vazbu na vstupní tok v DFD
- W) Pokud je volitelný use case diagram použit, tak každý funkční požadavek má vazbu na alespoň jeden use case, který má vazbu na aktéra
- X) Každý use case, který má vazbu na aktéra, má vazbu i na alespoň jeden funkční požadavek
- Y) Use case má alespoň jednu vazbu na aktéra nebo je cílem alespoň jedné dependency nebo obojí

Příklady typických obecných znaků nekonsistentnosti, jimž je třeba věnovat v tomto kroku pozornost, jsou:

- identifikované události, nezahnuté do žádné reakce,
- evidentně existující výstupy systému (reakce), nevázané na žádnou událost,
- reakce na jedinou událost (to je velmi netypické, smysl procesům dávají až kombinace různých událostí v reakci, pravděpodobně tedy nebyly dostatečně analyzovány všechny souvislosti událostí [není-li odůvodněno a vysvětleno jinak, například nutnou globálností popisu]),
- detailní souvislosti událostí různých procesů, jimž neodpovídají vazby mezi těmito procesy,
- proces bez výstupů,
- proces bez vstupů,
- proces bez aktérů,
- strukturální souvislosti vnitřních a vnějších prvků procesů:
 - události či reakce, jež nejsou vzájemně alternativní, vázané k alternativním větvím procesu,
 - povinné součásti téhož výstupu procesu, vázané k jeho alternativním větvím,
 - obdobné nekonsistence v opakujících se částech procesů, neodpovídajících příslušné mocnosti vztahu částí výstupů či událostí apod.,
- apod.

Výstupem kroku jsou úpravy již vytvořeného **systému identifikovaných elementárních procesů** ve smyslu a s cílem zajištění pokud možno úplné konsistence různých modelů, resp. částí modelů.

- je „optimální“ ve smyslu ekonomickém (efektivní) i věcném (maximálně jednoduchý při zachování úplné funkčnosti);
- umožní následnou optimalizaci, implementaci a zavedení systému procesů, které respektují výše uvedené charakteristiky.

Základem této metodiky je technika analýzy událostí.

Cílem techniky je **identifikovat základní procesy v organizaci**. Události zde slouží k identifikaci příslušných řetězců činností – procesů. Analýza událostí je tak pomůckou k analýze jednotlivých činností, která tvoří podstatu procesní analýzy organizace – v tom smyslu, že zjišťuje, které činnosti patří „objektivně“ dohromady (tvoří tzv. „konceptuální proces“).

Základním východiskem metodiky je teze, že **činnost organizace**, coby soustavu jednotlivých procesů, je **modelem cílů organizace, skutečnosti, ovlivňujících splnění těchto cílů a jejich vzájemných souvislostí**. Z předchozího vyplývá, že veškeré činnosti v organizaci prováděné a jejich souvislosti musí sloužit výhradně cílům organizace a respektovat výše uvedené okolnosti.

Metodika analýzy procesů se řídí třemi základními, vzájemně souvisejícími principy, postavenými na této tezi (předpokladu):

1. **Princip modelování** vyjadřuje předpoklad, že **objektivním základem** k implementaci podnikových procesů musí být **reálné skutečnosti**, existující **vně a nezávisle** na organizaci.
2. **Princip různých architektur procesu** vyjadřuje potřebu oddělit od sebe při konstrukci/poznávání procesů takové charakteristiky procesu, které jsou dány objektivními, na organizaci **nezávislými skutečnostmi** (konceptuální charakteristiky procesu) od charakteristik, daných konkrétním **kontextem umístění procesu v organizaci** (implementačních charakteristik procesu).
3. **Princip abstrakce** vyjadřuje způsob, jakým jsou jednotlivé identifikované skutečnosti **podrobněji analyzovány prostřednictvím hierarchických abstrakcí**.

Jednotlivé výše zmiňované principy jsou, co do svých důvodů a příslušných důsledků, analogickou aplikací principů, běžně respektovaných (téměř zřejšími) metodikami vývoje informačního systému.

Následuje stručný popis postupu podle této metodiky.

9. Metodický postup modelování podnikových procesů

Základem formulace procesů v organizaci jsou:

- identifikované **základní činnosti** (úkony případných procesů),
- představa o základních událostech a předpokládaných reakcích na ně (kontextová představa organizace),
- představa o základních objektech zájmu a jejich životních cyklech (objektová představa organizace).

Samotná analýza procesů probíhá ve třech fázích:

1. **Analýza elementárních procesů**, jejímž výsledkem jsou zjištěné elementární procesy, jejich struktura a vzájemné vazby, a to na základě analýzy událostí a reakcí a jejich vzájemných souvislostí.
2. **Specifikace klíčových procesů**, jejímž výsledkem jsou zjištěné klíčové procesy v organizaci, jejich struktura, vzájemné vazby a jejich podstatné atributy, a to na základě objektové analýzy produktů organizace společně s výsledkem předchozí fáze – zjištěnými elementárními procesy, z nichž se klíčové procesy skládají.
3. **Specifikace podpůrných procesů**, jejímž výsledkem jsou zjištěné podpůrné procesy v organizaci, jejich struktura, vzájemné vazby a jejich podstatné atributy, a to na základě objektové analýzy organizace společně s výsledkem předchozích fází – zjištěnými elementárními a klíčovými procesy.

Po analýze procesů, jejímž výsledkem je konceptuální procesní model organizace, se předpokládá fáze **implementace procesů**, kde se jednotlivé procesy transformují do konkrétní podoby, zohledňující konkrétní implementační specifika (specifika organizační a technologické infrastruktury organizace). Implementační model procesů je poslední (úrovní modelu procesů a je podkladem k dalším navazujícím činnostem zavedení systému procesů (tj. vytvoření příslušných organizačních a technických podmínek pro běh procesů, naplňování a následné provedení projektu zavedení systému procesů). Jako součást postupu ještě před fází implementace procesů lze počítat i případný reengineering podnikových procesů.

Základní procesní charakteristiky postupu (následnost, variantnost a provádění jednotlivých fází a kroků) ilustruje obrázek 9.1.

9.1 Jednotlivé fáze postupu analýzy podrobněji

Jak ilustruje obrázek 9.1, sestává postup analýzy procesů ze tří paralelně probíhajících a vzájemně koordinovaných fází, do nichž jsou zařazeny jednotlivé kroky s tím, že celý postup začíná samostatným krokem **Analýza událostí a vnějších reakcí**, jehož výsledkem je globální východiskem veškerého následného postupu.

0. krok – Analýza událostí a vnějších reakcí

Cílem tohoto kroku je zjistit veškeré relevantní reálné události, které vedou, nebo jsou podstatné pro dosažení cíle, vznik produktů a provádění činností podnikových procesů, a tyto události přiřadit vnějším reakcím. Hovoří se zde o **vnějších reakcích** s cílem zdůraznit, že podstatné jsou ty reakce, které **směřují mimo organizaci**. To též platí i o událostech – za **události** jsou zde považovány takové, které **vznikají mimo organizaci**, jediné ty jsou totiž plnohodnotnými reprezentanty skutečné objektivních důvodů k činnostem podniku.