

35. Konceptuální funkční modely používané při analýze

Funkční model

- notace
- model jednání (seznam událostí + scénáře)
- pokud scénář obsahuje složitější aktivity, pak dekompozice těchto aktivit na jednodušší (scénáře, diagramy aktivit, hierarchická sada DFD - kontextový diagram + diagramy úrovně 0,1, ... + popis
- minispecifikace elementárních operací
- datový slovník

Model jednání, případy použití (use cases)

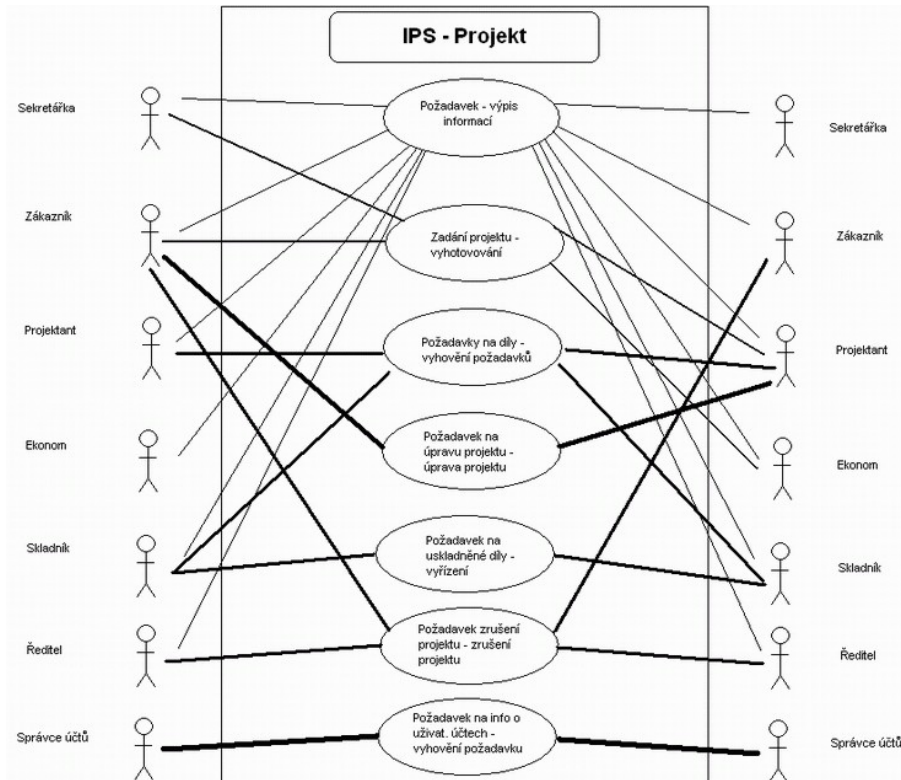
Model jednání (Use Case Diagram) vyjadřuje vztahy mezi Actors vně systému a Use Case uvnitř systému. Tato část UML nejméně rozpracovaná část celé metodiky, lze v ní snad nalézt určitou analogii s DFD nejvyšší úrovně.

Use cases viz. otázka 33 – UML

- mimo jiné scénáře událostí
- popis uživatelského vzhledu
- objektový model domén

Model jednání obsahuje:

- Aktér – uživatelská role nebo spolupracující systém
- Hranice systému – vymezení hranice systému
- Případ použití – dokumentace události, na kterou musí systém reagovat
- Komunikace – vazba mezi aktérem a případem použití



Ilustrace 1: Model jednání

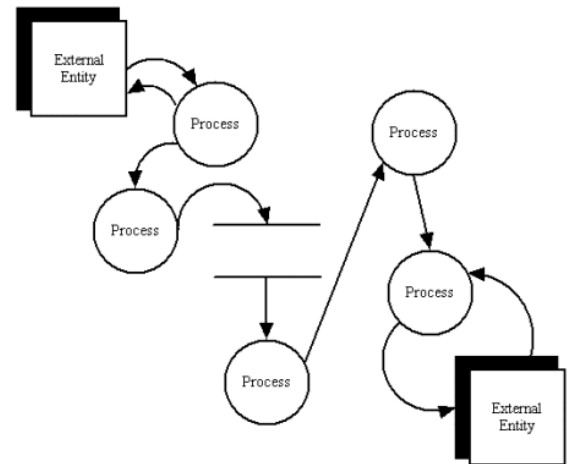
- interakce systému a uživatele
- události (akce, které vyvolali reakci – vstup)
- reakce (odezvy na události – výstupy)
- časová osa

- popisy událostí
 - například událost A

POPIS AKCE	POPIS OPERACE	NÁZEV
READS	Jaká data čte	
OPERATION	Vložení dat zboží do databáze	
DESCRIPTION	Uživatel systému správně vyplní formulář pro vkládání informací o zboží do databáze	
CHANGES	Jaká data mění	
SENDS	Jaká reakce vyvolává	
ASSUMES	Co předpokládá	
RESULT	Co zjišťuje	

Diagramy datových toků (DFD), datové toky a datové paměti

- Data Flow Diagram
- systém chápán jako množina dat
- zachycení vazeb funkcí a toků dat (vstupních i výstupních)
- komponenty:
 - proces
 - datové toky (orientované hrany)
 - datové paměti (místa, kde si potřebujeme něco pomatovat)
 - aktéři (uživatelské role nebo spolupracující systémy)
 - externí entita (zdroje a destinace dat)

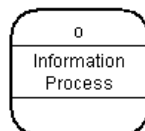


Existují 2 možné notace: Yourdon & Coad a Gane & Sarson

Proces transformuje vstupní data na výstupní



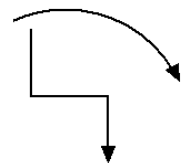
Yourdon & Coad



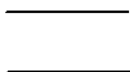
Gane & Sarson

Ilustrace 2: Ukázka DFD diagramu

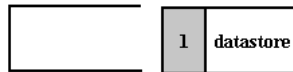
Datové toky jsou roury, kterými putují pakety informací. Obvykle se užívá značení šipkou společně se jménem. Oba modely obvykle užívají shodnou notaci, i když Gane & Sarson inklinují k pravoúhlým hranám.



Datové paměti jsou repozitáře dat v systému. Občas jsou připodobňovány k souborům.

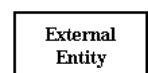
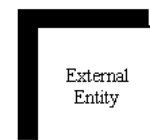


Yourdon & Coad



Gane & Sarson

Externí entita jsou objekty mimo systém, se kterými systém komunikuje. Mohou být buď zdroji, nebo destinacemi dat. Notace je opět nepatrně odlišná u obou pro oba typy.



Vrstvy DFD

DFD se standardně navrhuje v několika úrovních. Nejprve se navrhne kontextový diagram, následovaný několika vrstvami DFD diagramů

- kontextový diagram
- DFD systému
- DFD pro každý proces
- další podrobnější vrstvy DFDů

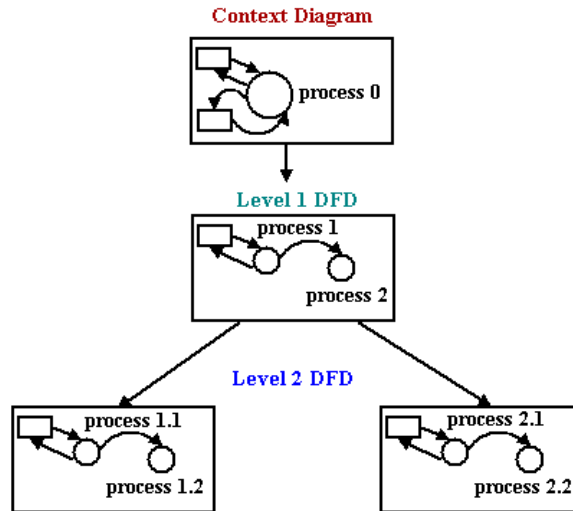
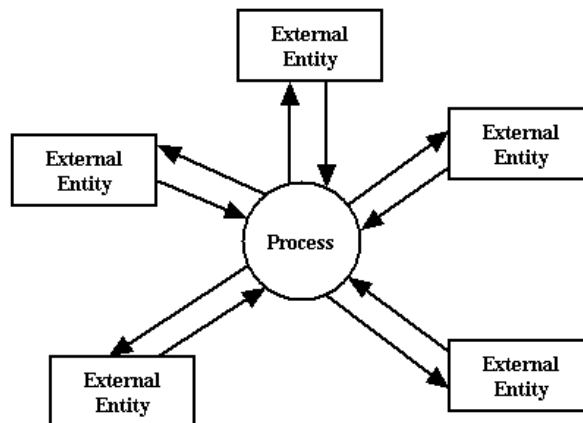


Diagram kontextu

Kontextový diagram je „top level“ (známý také jako level 0) *Data Flow Diagram*. Obsahuje pouze jeden proces (proces 0), který představuje funkčnost celého systému ve vztahu k externím entitám. Systém = „black-box“. Ukázka:



Minispecifikace

- popisuje proces na nejnižší úrovni
- lze snadno převést do programovacího jazyka
- nemusí obsahovat věty
- může obsahovat abstraktní programovací kód (`if <uživatel je zaregistrovaný> then <ukaž seznam>`)

Příklad podle X36SIN:

1. Zákazník prohlíží katalog a vybere si zboží k nákupu
2. Zákazník zvolí nákup
3. Zákazník vyplní dodací informace (adresa, expresní nebo standardní dodávka)
4. Systém zobrazí plnou cenu včetně ceny dodání

5. Zákazník vyplní platební informace (číslo kreditní karty)
6. Systém autorizuje platbu
7. Systém potvrdí prodej
8. Systém zašle potvrzovací e-mail zákazníkovi

Jiný příklad:

```
IF <odběratel je registrován> THEN
    IF <objednávka je správně vyplněna> AND <všechny druhy výrobků v
        objednávce existují> THEN
        přidej údaje do tabulky objednávka od odběratele.
    OTHERWISE
        informuj odběratele o chybě v objednávce.
OTHERWISE
    informuj odběratele o jeho neexistenci v databázi odběratelů.
```

- lze používat i nestandardní příkazy jako „for every“