Strukturovaná analýza a návrh

Yordonova moderní strukturovaná analýza(YMSA)

Strukturovaný návrh

Yourdonova strukturovaná analýza

- Esenciální model
- Implementační model

Části Esenciálního modelu

- Model okolí
- Prvotní model chování
- Dokončení esenciálního modelu

Model okolí

Definuje hranice mezi okolím systému a implementačním prostředím

Části modelu:

- Popis účelu systému
- Kontextový diagram
- Seznam událostí

Popis účelu systému

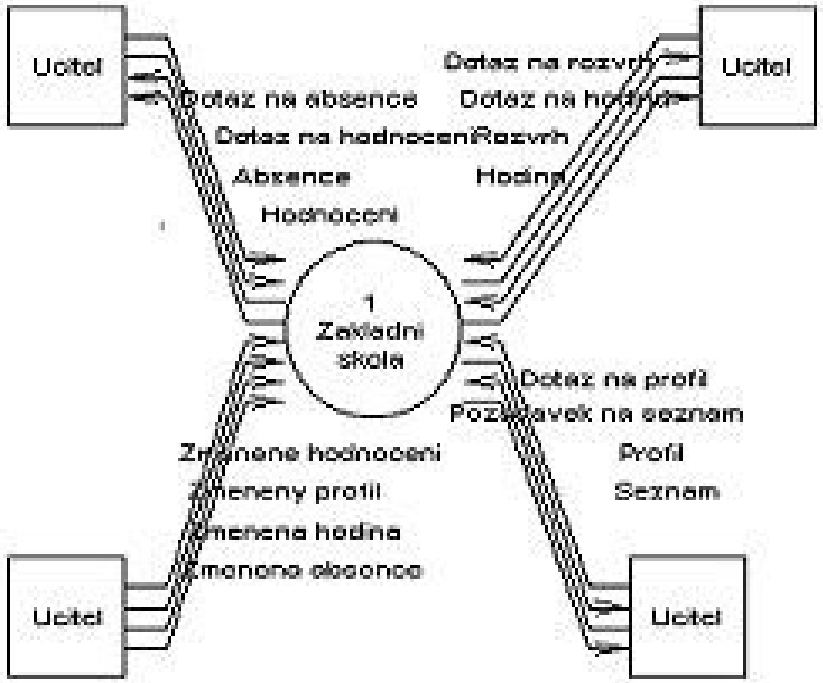
- *Textový dokument, který vytyčuje hlavní cíle celého vývoje systému*
- *Určen pro zakazníka*

Příklad:

Účelem vývoje Informačního systému základní školy je vytvořit účinný nástroj pro vedení agendy jakékoli základní školy. Ředitelství školy se může dotazovat na veškeré informace obsažené v systému, vytváří rozvrhy, přiřazuje učitele k jednotlivým třídám a předmětům a registruje osoby, které mohou přistupovat do systému. Každý pátek ve stejnou dobu ředitelství obdrží přehled absencí všech žáků ve škole. ...

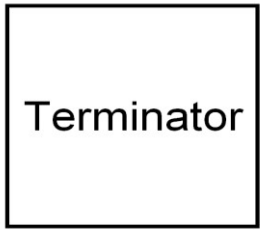
Kontextový diagram

- Speciální případ DFD (Data flow diagram) s jedním procesem, který reprezentuje celý systém
- Systém komunikuje s okolním světem přes terminátory



Terminátory

- Terminátor s mnoha vstupy a výstupy lze nakreslit vícekrát
- Terminátor komunikuje pouze se systémem, není možná komunikace mezi dvěma terminátory
- Název se volí podle role terminátoru
- Komunikace je reprezentována orientovanými šipkami



Vstup = požadavek (šipka z terminátoru)

Výstup = požadovaná data (šipka do terminátoru)

Seznam událostí

- Výčet podmětů z okolí, na něž musí systém reagovat
- Každá událost je označena písmenem
  - F(flow-oriented) = obsahuje reálná data
  - C(control) = impuls k provedení nějaké činnosti v závislosti na nějaké události
  - T(time) = časový impuls pro spuštění nějaké činnosti v určitém čase

– ***Příklad:***

Učitel se dotazuje na hodnocení (F)

Ředitel požaduje absence každý pátek(T)

Konstrukce modelu okolí

- nasbírat požadavky a představy uživatelů
- ze získaných informací vytvořit model okolí a seznam událostí

Každý uživatel má většinou představu jen o určité části systému, ve které pracuje.

Určen pro zákazníka, k vytvoření představy o vyvíjeném systému.

Může obsahovat i počáteční fázi E-R diagramu a datový slovník.

Konstrukce modelu okolí 2

- **Jestliže není jasné, s čím bude systém muset komunikovat, je vhodné začít s E-R diagramem.**
- **ERD ukáže objekty a vztahy mezi nimi.**
- **Podle vnějších událostí týkajících se dat v ERD se vytvoří kontextový diagram**

- **Kontrolní otázky**:
  - Každý vstupní tok na kontextovém diagramu je nezbytný pro rozpoznání události, 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 či změnit stav systému

Model chování

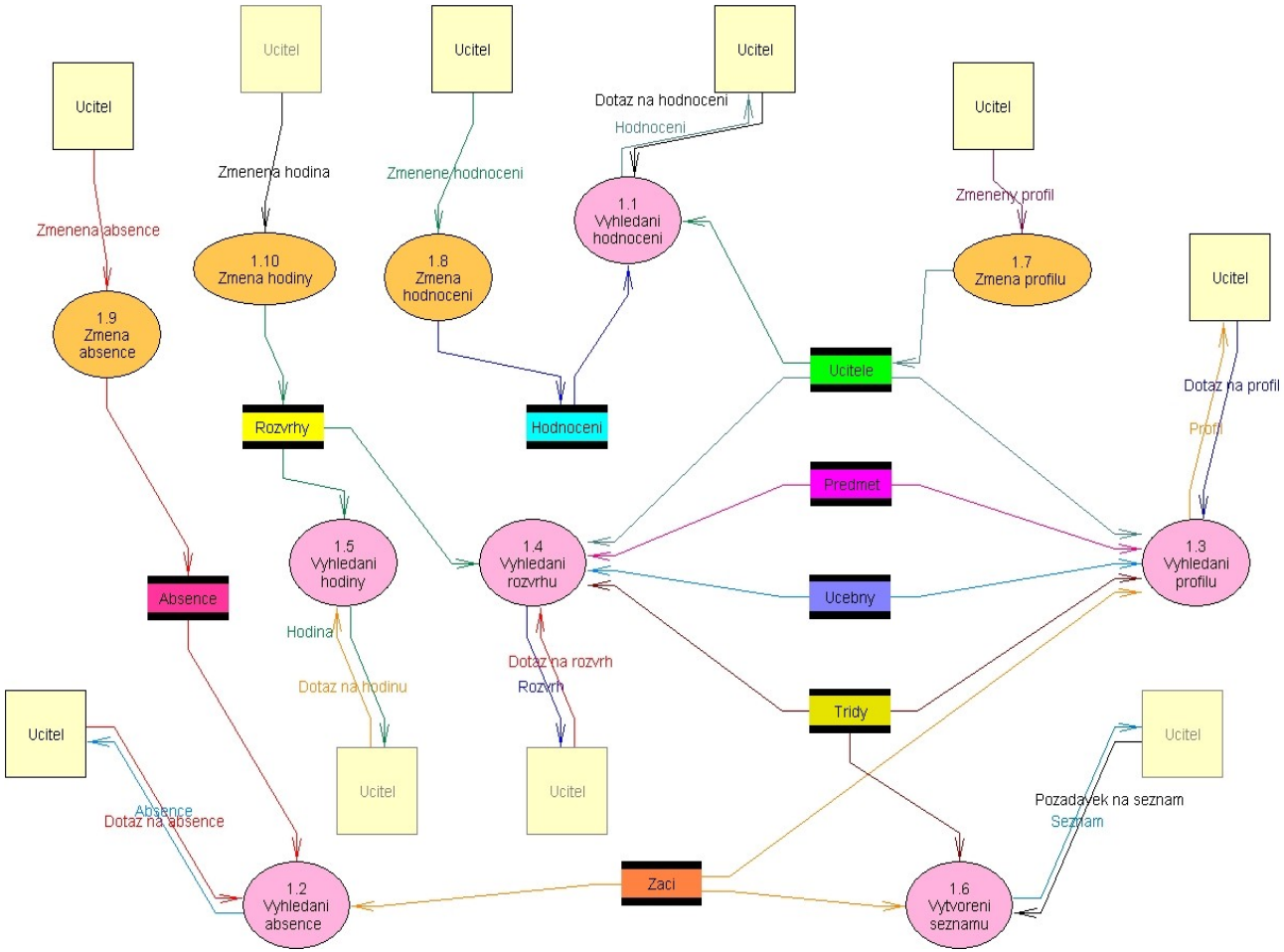
- **Popisuje chování uvnitř systému**

- **Části modelu:**
  - Prvotní model chování
  - Dokončení modelu chování

Prvotní model chování

- vychází z kontextového diagramu
- dekomponuje systém na dílčí procesy a definuje datové sklady.
- postupuje se „zdola-nahoru“:
  - pro každou událost v seznamu událostí nakreslíme proces
  - název procesu vystihuje reakci na událost
  - proces má zakresleny vstupní a výstupní toky a paměti tak, aby proces byl schopen vytvořit požadovanou reakci
- souběžně s DFD se modeluje i ERD
- funkce řídících toků a procesů se modelují v STD
- po dokončení se porovná s kontextovým diagramem a seznamem událostí
- není určen zákazníkovi

Prvotní model chování



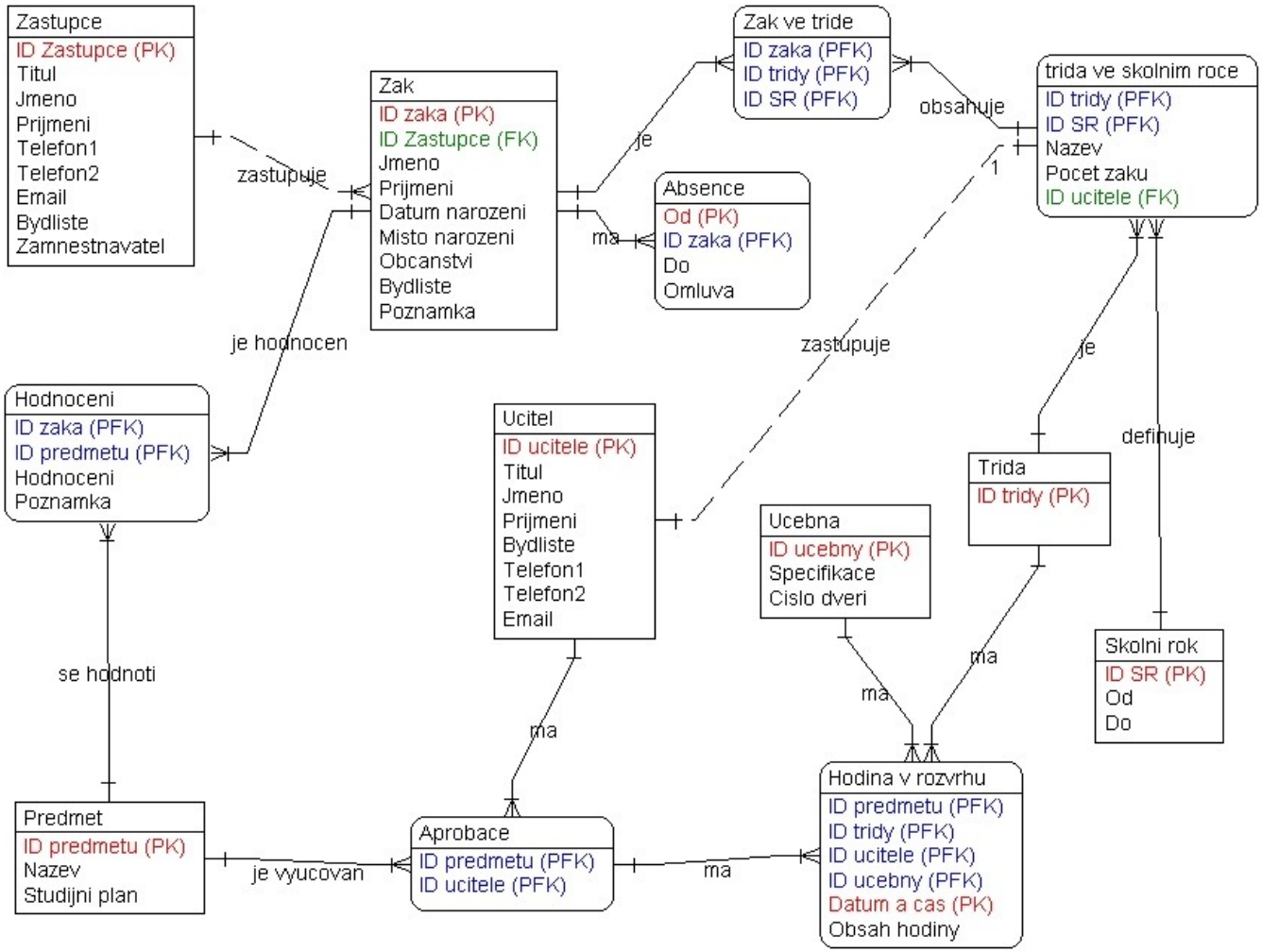
Dokončení modelu chování

- z prvotního modelu chování vytvoříme procesy na vyšší úrovni, ve kterých budou příbuzné procesy skryty pod jeden proces
- každá skupina procesů sjednocena pod jeden proces musí mít podobné odpovědi, tudíž týkající se jedné záležitosti
- paměťse skryje do společného procesu také pokud ji nepoužívá žádný proces vně nadřazeného procesu
- Pravidlo „ 7±2“
- v případě nezřetelné funkční dekompozice, lze odvodit podle vstupních a výstupních toků

Dokončení modelu chování 2

- žádná položka v datovém slovníku si nesmí vzájemně odporovat s jinou položkou, a zároveň se musí zkontrolovat se všemi úrovněmi DFD, ERD a minispecifikaci.
- minispecifikace procesů na nejnižší úrovni se píše až po vytvoření předběžného modelu chování
- ERD je vyvíjen v návaznosti na DF
- v případě závislosti události na čase se vyvýjí STD
- Vždy je nutné vyvažování !!

ERD


Zastupce
ID Zastupce (PK)
Titul
Jmeno
Prijmeni
Telefon1
Telefon2
Email
Bydliste
Zamnestnavatel
zastupuje
Zak
ID zaka (PK)
ID Zastupce (FK)
Jmeno
Prijmeni
Datum narozeni
Misto narozeni
Obcanstvi
Bydliste
Poznamka
je
Zak ve tride
ID zaka (PFK)
ID tridy (PFK)
ID SR (PFK)
obsahuje
trida ve skolnim roce
ID tridy (PFK)
ID SR (PFK)
Nazev
Pocet zaku
ID ucitele (FK)
1
ma
Absence
Od (PK)
ID zaka (PFK)
Do
Omluva
je hodnocen
zastupuje
Hodnoceni
ID zaka (PFK)
ID predmetu (PFK)
Hodnoceni
Poznamka
Ucitel
ID ucitele (PK)
Titul
Jmeno
Prijmeni
Bydliste
Telefon1
Telefon2
Email
je
definuje
Trida
ID tridy (PK)
Ucebna
ID ucebny (PK)
Specifikace
Cislo dveri
Skolni rok
ID SR (PK)
Od
Do
se hodnoti
ma
ma
ma
Predmet
ID predmetu (PK)
Nazev
Studijni plan
je vyucovan
Aprobace
ID predmetu (PFK)
ID ucitele (PFK)
ma
Hodina v rozvrhu
ID predmetu (PFK)
ID tridy (PFK)
ID ucitele (PFK)
ID ucebny (PFK)
Datum a cas (PK)
Obsah hodiny

Datový slovník

- ***Dotaz na hodnoceni*** *= @ ID zaka + @ID predmetu* ***Hodnoceni*** *= @ ID zaka + @ID predmetu + Hodnoceni + (Poznamka)* ***Dotaz na absence*** *= @Od + @ID zaka*
- ***Absence*** *= @Od + @ID zaka + (Do) + (Omluva)*
- ***Dotaz na profil*** *= @ID profilu*
- ***ID profilu*** *= [@ID zaka | @ID Zastupce | @ID Predmetu | @ID Ucebny | @ID Tridy]*
- ***Profil*** *= [Ucitel | Zak | Zastupce | Predmet | Ucebna | Trida]*
- ***Ucitel*** *= @ID ucitele + (Titul) + Jmeno + Prijmeni + Telefon1 + (Telefon2) + Email + Ulice + Mesto*
- ***Zak*** *= @ID zaka + @ ID zastupce + Jmeno + Prijmeni + Datom narozeni + Misto narozeni + Obcanstvi + Ulice + Mesto + Poznamka*
- ***Zastupce*** *= ID zastupce + (Titul) + Jmeno + Prijmeni + Telefon1*
- *+ (Telefon 2) + Email + Zamnestnavatel + Ulice + Mesto*
- ***Predmet*** *= @ID predmetu + Nazev + Studijni plan*
- ***Ucebna*** *= @ID ucebny + Specifikace + Cislo dveri*
- ***Trida*** *= @ID tridy + @ID SR + @ID ucitele + Nazev + Pocet zaku*
- ***Dotaz na rozvrh*** *= @ID predmet + @ID tridy + @ID ucitele +@ID ucebny*
- ***Rozvrh*** *= @ID predmetu + @ID tridy + @ID ucitele + @ID ucebny + Datum a cas + Obsah hodiny*
- ***Dotaz na hodinu*** *= @ID predmetu + @ID tridy + @ID ucitele + @ID ucebny*
- ***Hodina*** *= Obsah hodiny*
- ***Dotaz na seznam zaku*** *= @ID tridy*
- ***Seznam zaku ve tride*** *= {@ID zaka + Jmeno + Prijmeni}*
- ***Zmeneny profil =*** *@ID ucitele* ***+*** *(Titul) + (Jmeno) + (Prijmeni) + (Telefon1) + (Telefon2) + (Email) + (Ulice) + (Mesto)*
- ***Zmenene hodnoceni*** *= @ID zaka + @ID predmetu + Hodnoceni + (Poznamka)*
- ***Zmenena absence*** *= @ID zaka + @Od + (Do) + (Omluva)*

Implementační model

- poslední část metodiky YMSA
- na vývoji se podílí analytici, návrháři a programátoři
- vymezuje části systému vykonávané automaticky a manuálně
- úkoly:
  - vymezení hranic
  - vymezení uživatelského rozhraní
  - manuální podpora
  - stanovení provozních omezení

Minispecifikace

Vyhledání hodnocení

GET vstup FROM Ucitel, Reditelstvi, Zastupce, Tridni ucitel.
IF ID Zaka AND ID Predmetu, THEN
    READ FROM Hodnoceni.
SELECT ID Zaka, ID Predmetu, hodnoceni, poznamka.
SEND vystup TO Ucitel, Reditelstvi, Zastupce, Tridni ucitel.

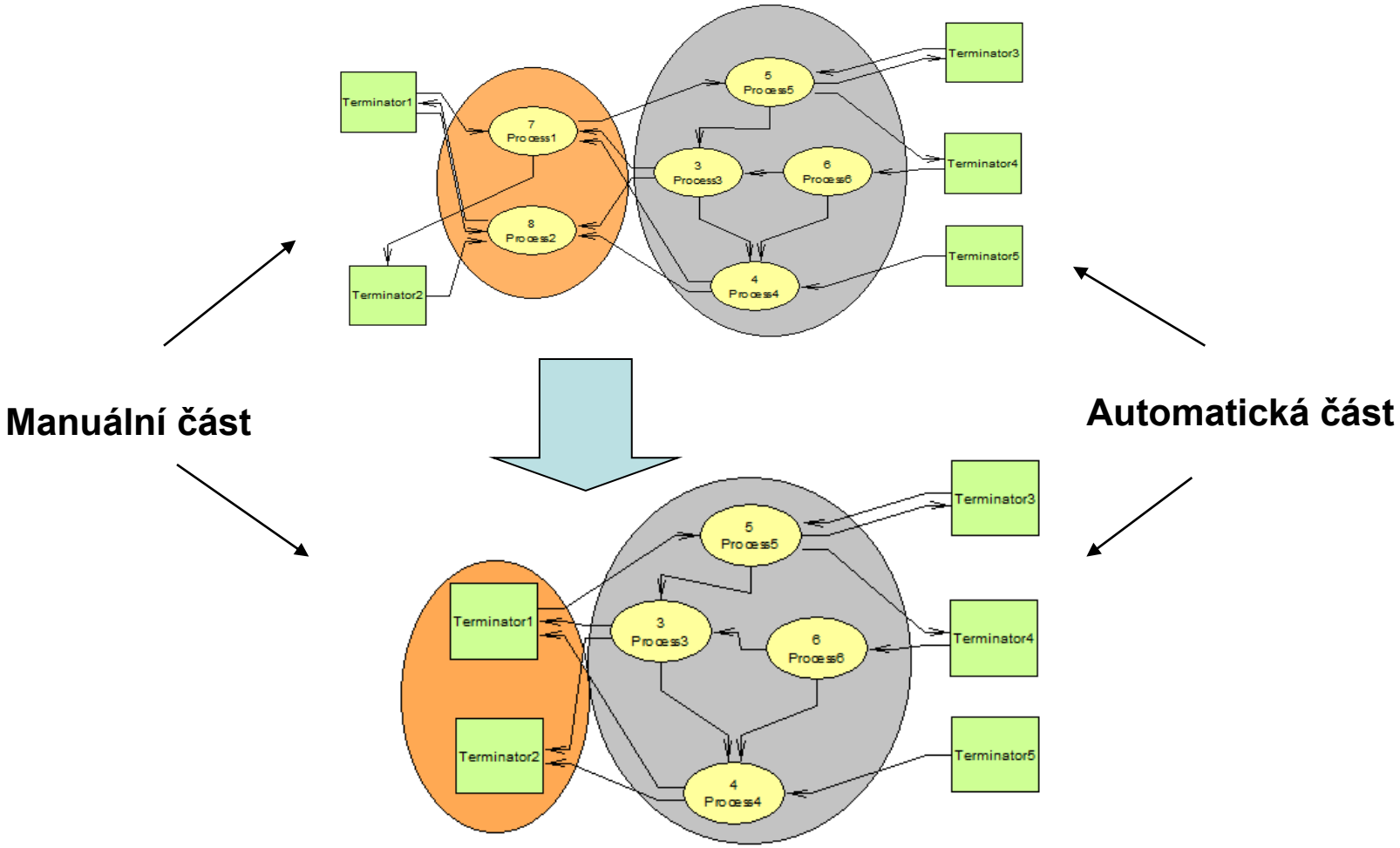
Změna hodnocení

GET vstup FROM Ucitel.
IF ID zaka AND ID predmetu,
  THEN WRITE hodnoceni TO Hodnoceni.
OTHERWISE “Chybné zadání”

Vyhledání rozvrhu

GET vstup FROM Ucitel, Reditelstvi, Zastupce.
SELECT
    CASE 1 (ID tridy):
        READ FROM Tridy.
SELECT rozvrh tridy.
    CASE 2 (ID predmetu):
        READ FROM Predmety.
SELECT rozvrh predmetu.
    CASE 3 (ID ucebny):
        READ FROM Ucebny.
SELECT rozvrh ucebny.
    CASE 4 (ID ucitele):
        READ FROM Ucitele.
        SELECT rozvrh ucitele.
SEND vystup TO Ucitel, Reditelstvi, Zastupce.

Implementační model 2



Omezení v implementačním modelu

- velikost paměti, rychlost přesunu a objem přenesených dat z paměti i do paměti.
- časová odezva na vstup
- zákazník trvá na užití konkrétního hardwaru nebo programovacího jazyka apod.
- speciální nároky prostředí, ve kterém bude systém provozován. (teplota, vlhkost, spotřeba energie, hluk,…)
- spolehlivost a dostupnost
- práva a oprávnění pro jednotlivé uživatele

Strukturovaný návrh

- **navazuje na práci analýzi**
- **čerpá z DFD**
- **hledá posloupnosti v práci procesů**
- **podle funkce se procesy dělí na:**
  - transformační (transformují vstupy na výstupy)
  - transakční (převádí data dalším procesům podle obsahu)
- **návrh datových toků používá dvě základní techniky analýzy:**
  - transformační
  - transakční

Transformační analýza

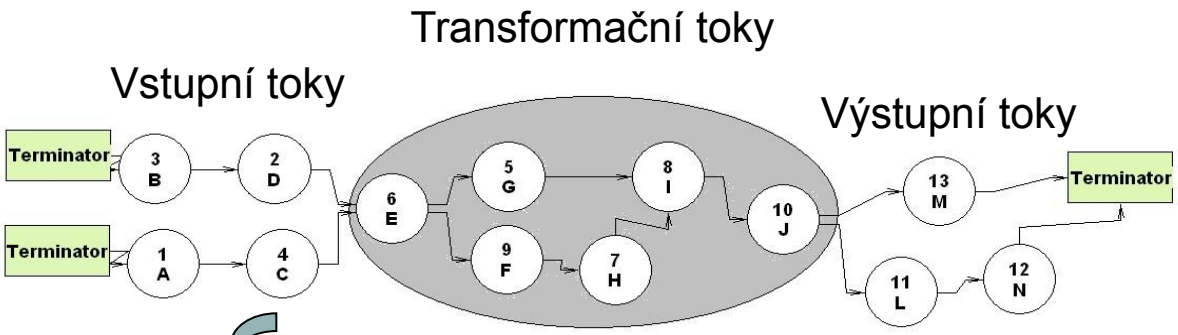
1. v DFD se postupuje od terminátoru po vstupních tocích směrem do vnitřních částí modelu do té doby, než data, která po těchto tocích plynou, nejsou zpracována
2. tok před samotným zpracováním se označí značkou
3. po výstupních tocích se postupuje od terminátoru v protisměru toku dat do středu diagramu
4. ve chvíli kdy se narazí na procesy, která data zpracovávají a ne jen formátují pro výstup, se poslední tok před vstupem do procesu označí
5. všechny označené toky se spojí křivkou a vzniklá podmnožina grafu, tvoří centrální transformaci DFD.

Transformační analýza 2

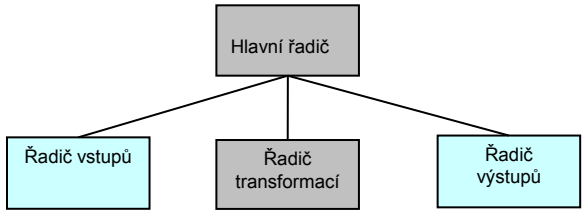
6. pokud podmnožinu grafu tvoří více jak jeden proces, přidá se další proces, přes který jsou přesměrovány všechny toky mezi procesy v podmnožině grafu a tento proces je prohlášen za centrální transformaci
7. kořen SSD označuje centrální proces upraveného DFD
8. Nejvíce vlevo od kořene se přidávají pouze procesy generující data pro centrální proces
9. nejvíce vpravo jsou přidávány procesy přijímající data od centrálního procesu
10. mezi centrálním procesem a procesy na okrajích stromu jsou procesy, jež přijímají i produkují data. Z těchto procesů se vytvoří bezprostřední následníci centrálního procesu
11. vazby mezi procesy reprezentují přenášená data.

- stejným způsobem pokračujeme i na nižších úrovních

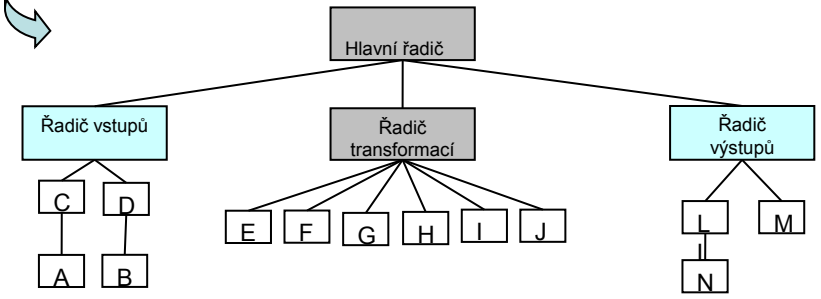
Transformační analýza 3



1.úroveň faktorizace:



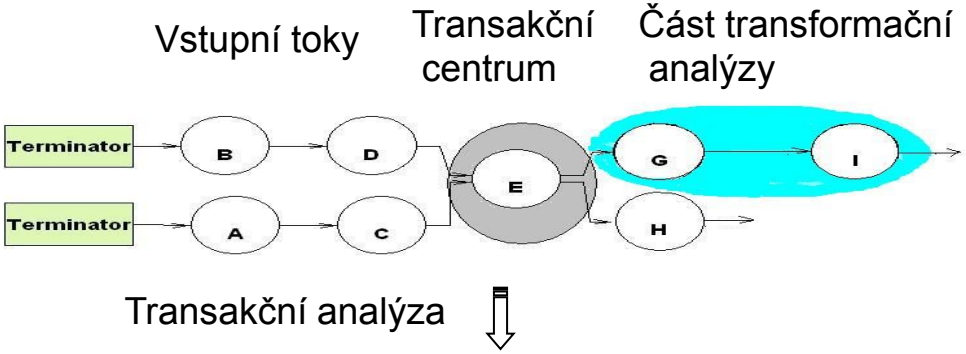
2.úroveň faktorizace:



Transakční analýza

- v případě výskytu transakčních toků
  1. najde se proces sloužící jako výchozí bod akčních cest, ve struktuře systému se určí jako transakční centrum
  2. podgrafy, jež vzniknou v DFD vymezením transakčních center, se zobrazí do SSD jako následníci transakčního centra
  3. struktura podgrafů je zakreslena do celkového SSD spolu s transformačními moduly, podle způsob zpracování dat, s použitím transakční nebo transformační analýzy

Transakční analýza 2



Optimalizace podle návrhových heuristik

- vyhodnotit primární programové struktury, jejímž cílem je redukce propojení a zvýšení soudržnosti
- minimalizovat rozšiřující se struktury a pokusit se přiblížit větve se vzrůstající hloubkou
- udržet obsah modulu v rozsahu, který je vymezen řízením datového modulu
- snižovat složitost a nadbytečnost modulových rozhraní a zvýšit soudržnost modulů
- definovat moduly s jasnou funkcí a vyhnout se příliš restriktivním modelům
- vyhledat moduly s jediným vstupem a výstupem a vyhnout se pochybným vazbám
- balit software s ohledem na návrhová omezení a požadavky na přenositelnost

Diagram struktury systému