

Strukturovaný přístup k analýze a návrhu IS

Klasickou metodou analýzy a návrhu informačních systémů je **strukturovaný přístup**, navržený v 70. letech (Tom DeMarco, Ken Orr, Larry Constantine, Vaughn Frick, Ed Yourdon, Steven Ward, Peter Chen...).

Základy strukturovaného přístupu položil Tom deMarco v roce 1979 v práci „Strukturovaná analýza a specifikace systémů“. Mezi jeho základní doporučení patří:

- **rozdělit systém na subsystemy;**
- **používat grafické znázornění (grafické modely) systému;**
- **před implementací vytvořit logický model systému.**

Strukturovaný přístup modeluje systém pomocí následujících prostředků:

1. ERD – Entity Relationship Diagram
2. DFD – Data Flow Diagram
3. FSD – Function Structure Diagram
4. STD – State Diagram
5. DD – Data Dictionary
6. Structure Chart
7. Flow Diagram

Úrovně z hlediska abstrakce

- **Konceptuální model** – rozpoznání základních datových objektů a jejich vztahů –návrh – co je obsahem systému, návrh je nezávislý na technologickém prostředí
- **Logický model** – relační schéma (včetně integritních omezení) – určuje jakou mají data strukturu, není zatížen konkrétní implementací. Často vyjádřen formou ERD a DFD diagramů.
- **Fyzický datový model** – implementace v konkrétním databázovém produktu

ANALÝZA – DESIGN – IMPLEMENTACE

Strukturovaný přístup – data a funkce

Strukturovaný přístup k analýze a návrhu pohlíží na informační systém ze dvou různých úhlů pohledu:

– z pohledu **dat**

– z pohledu **funkcí**

Každý má svoji specifickou logiku (ERD x DFD).

Abstrakce **část - celek (agregace)**. Tato abstrakce se typicky používá ve funkčním modelu, kde se dělí systém na subsystémy, části subsystémů atd. Pro agregaci je typická principiální neomezenost dělení.

Abstrakce **specifický podtyp - obecný typ (generalizace)**. Tato abstrakce se typicky používá v datovém modelu, kde je možné uvažovat o jednotlivých specifických variantách nadřazeného pojmu (entity, objektu). Na rozdíl od agregace není nadřazený celek definován jako souhrn podřazených částí, ale jako nositel jejich společných vlastností (atributů).

Je důležité, že tyto dva základní typy abstrakce jsou vzájemně neslučitelné a tím tvoří jádro základního rozporu mezi funkčním a datovým modelem.

Objektově orientované metody se pokoušejí tento rozpor překonat zapouzdřením obou - dat i funkcí - do jediného objektu.

ERD (Entity Relationship Diagram)

Peter Chen, 1976, <http://bit.csc.lsu.edu/~chen/chen.html>

ER diagramy slouží k modelování **struktury** informačního systému a databáze v grafické podobě.

Cílem ERD je zmapovat data ukládaná do databáze a jejich vzájemné vztahy.

Výstupem ERD je popis logické struktury databáze:

- **Entita** – objekt, který je předmětem zájmu
- **Atribut** – elementární datový prvek, který entitu blíže charakterizuje
- **Relace** – vztah mezi dvěma entitami
- **Kardinalita vztahu** – mocnost vztahu mezi entitami: 1:1, 1:N, N:1, M:N
- **Parcialita vztahu** – povinnost nebo volitelnost vztahu

Vztah je informace, kterou si systém musí pamatovat, nelze ji odvodit.

Je rozdíl mezi ERD (Chen) a relačním datovým modelem (Codd), tj. je rozdíl mezi pojmy Relationship (vztah) a Relation (relace, která odpovídá v databázi tabulce) – viz dále vysvětlení modelů.

ER model popisuje **statický pohled na systém**.

DFD (Data Flow Diagram)

Cílem DFD je modelování **datových nebo řídicích toků v systému** (v grafické podobě).

DFD diagramy popisují funkce systému.

U DFD diagramů existuje několik forem záznamu, nejčastěji se používá notace dle DeMarca nebo Yourdana.

DFD se skládá z prvků „**proces**“, „**datový sklad**“, „**terminátor**“. Datovým skladem v tomto pojetí rozumíme jakékoli úložiště dat.

DFD diagramy mají **hierarchickou úroveň**, procesy se dají postupně zjemňovat. Na nejvyšší úrovni stojí **kontextový diagram** – jedná se o speciální případ DFD diagramu, kdy je systém zobrazen jako jediný proces a naznačuje vztah systému s okolím. Na dalších úrovních jsou popisovány jednotlivé procesy a datové nebo řídicí toky.

Dekompozice DFD na nižší úrovně až na úroveň základních elementárních funkcí:

- Nesmí existovat proces, který nemá žádné vstupy a přesto produkuje datové toky
- Nesmí existovat proces, který pouze spotřebovává data a nemá výstupy

FSD (Function State Diagram)

Účelem FSD je zobrazit dekompozici systému na funkční celky (subsystémy), zdokumentovat funkční hierarchii systému a poskytnout pohled na vyvíjený systém se zaměřením na jeho hierarchickou strukturu.

FSD diagram se skládá z

- funkce (proces, systém)
- vazba

Funkce dělíme na:

- procesní
- dialogové
- řídicí

Rozdíl mezi DFD a FSD je následující:

- FSD je zaměřen na hierarchickou strukturu subsystémů, jde o statický pohled
- DFD je zaměřen na datové toky a procesy transformující vstup na výstup, jde o dynamický pohled na systém

STD (State Transition Diagram)

STD obsahuje sled stavů, v jakém se systém (nebo jeho část) může nacházet a za jakých podmínek může dojít ke změně stavu. Modeluje časově závislé chování systému.

- Důležitý z hlediska pochopení logiky systému
- Stavy jsou statické, změna stavu je většinou důsledek nějaké události.

- Vyjádření – např. vývojové diagramy
- Může být hierarchický.

STD diagram musí mít jeden počáteční bod a jeden nebo více koncových bodů. Samotný stav je vnímán jako statický. Diagram eviduje podmínku, při které systém přejde z jednoho stavu do druhého.

Data Dictionary (datový slovník)

Datový slovník slouží k formalizovanému popisu dat systému z pohledu uživatele.

Metadata (data o datech):

- Spravuje databázový stroj
- Přístup jen ke čtení

Vytvoříme-li datový slovník, který obsahuje všechny položky databáze, dostaneme informaci, zda se některé položky nevyskytují v databázi vícenásobně. Tím pádem můžeme upravit datový model a snížit redundanci.

Structure chart

Slouží pro znázornění hierarchie programových modulů systému, ve tvaru grafu (stromu). Kořenem je hlavní programový modul, uzly znázorňují dílčí volané moduly.

Flow chart (vývojový diagram)

Zachycuje algoritmus. Alternativní vyjádření algoritmu, vhodnější pro strukturované programování je strukturogram. Pro modelování systému má Flow Chart nejmenší důležitost, lze použít pro popisy funkcí.

Návrh IS

Výše uvedené diagramy slouží pro návrh informačního systému. Návrh může být dále doplněn **případy užití**, modelem spolupráce nebo funkčním modelem. Důležitá je také volba architektury, dnes převažuje architektura **třívrstvá** (prezentační, funkční a datová vrstva).

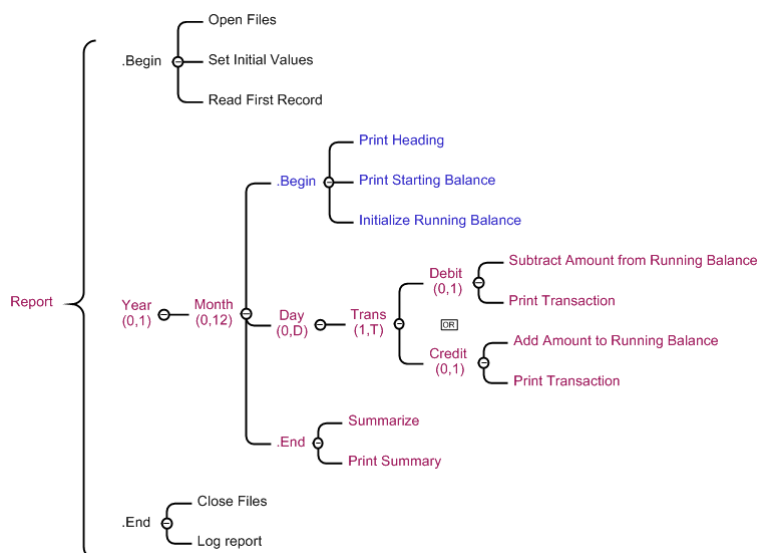
Přehled modelů

Logické modelování Gane – Sarson

Krátce po zveřejnění DeMarcovy "Strukturované analýzy a specifikace systému" (Structured System analysis: Tools and Techniques) publikovali Chris Gane a Trish Sarson svoji práci "Strukturovaná analýza systému". Metoda Gane/Sarson byla opět založená na **použití DFD jakožto výchozího modelu IS**, nově však zde bylo začleněno i datové modelování pomocí ERD.

Datově orientovaný přístup – Warnier/Orr

Vývoj této metody se datuje již od roku 1972. Jean Warnier (Francie) a Kenneth Orr (USA) svoji metodu založili na odvození logického datového modelu na základě **analýzy požadovaných výstupů systému**. Specifikují se také procesy výběru dat z databáze, transformace dat a výstupy. Model je rozšířen o stavové diagramy.



Ukázka Warnier-Orr diagramu – specifikace požadavků na report. [<http://www.mindapp.com/mind-map-examples/warnier-orr/report/>]

Entitně-relační model - Chen

Entitně vztahový model. Pracuje s pojmy **entita**, **atribut**, **vztah (relationship, association)**. Entita je množina objektů stejného typu. V objektovém modelování odpovídá pojmu „entita“ pojem „třída“. Atribut je datová složka entity. ERM nemusí vždy odpovídat relační implementaci. Při popisu vztahu v ERD diagramech by se vztah neměl zaměřovat za popis funkce.

Datový model

Zachycuje vztahy mezi daty, oproti konceptuálnímu modelu (Conceptual Model), který je odrazem reálného světa (a používá pro vyjádření ERD nebo prvky z objektového modelování...).

Relační datový model – Codd

Jedná se o vysoce abstraktní model určený pro modelování relačních databází. Pracuje s **pojmy relace (relation), atribut, funkční závislost**. V relačním datovém modelu nepracujeme s pojmy tabulka a řádek; pojmu „tabulka“ z fyzické realizace RDM v databázi zde odpovídá pojem „relace“. Mezi relacemi existuje funkční závislost. RDM by měl projít procesem **normalizace**, která vychází z koncepce funkčních vztahů. Cílem normalizace je navrhnout správné relace bez nadbytečných atributů (tj. bez redundance). Existuje celkem pět normálních forem (pátá je nejstriktnější), přičemž za správně navrženou databázi se považuje třetí normální forma. Na základě RDM lze pak vytvořit jeho fyzickou realizaci v konkrétním databázovém prostředí. Při fyzické realizaci pak podle skutečných potřeb aplikací může dojít k určité míře **denormalizace**, za účelem zrychlení reportovací vrstvy.

PERM – Physical ERM

ERM vztažený na prvky, které se skutečně vyskytují v relační databázi. Zde pracujeme s pojmy tabulka a řádek tabulky, který reprezentuje výskyt entity.

Yourdonova moderní strukturovaná analýza (YMSA)

Edward Yourdon vytvořil konceptuální popis systému, skládající se ze tří částí:

- a) Datový model
- b) Model chování
- c) Model řízení

Yourdon doporučuje pro modelování nástroj následujících vlastností:

- a) Musí být grafický
- b) Podpora TPODOWN přístupu
- c) Minimalizace redundance – řešení pomocí Data Dictionary
- d) Snadná čitelnost
- e) Předvídat chování systému [2]

Jádro metodiky (1989) spočívá v nalezení **esenciálního modelu**, který vyjadřuje podstatu systému, je dlouhodobě stabilní a je nezávislý na použité technologii a implementaci. Z esenciálního modelu je následně odvozen implementační model. [3]

Esenciální model se skládá ze dvou částí:

- a) **Model okolí**
- b) **Model chování**
- c) **Implementační model**

Model okolí se skládá z:

- a) Dokument o účelu systému
- b) Kontextový diagram
- c) Seznam událostí

V kontextovém diagramu je celý systém znázorněn jako jeden proces. Cílem je zachytit výměnu informací s okolím, tedy hlavně s uživateli systému.

Model okolí a zejména dokument o účelu systému je určen pro zákazníka a management. Model chování naproti tomu je vytvářen pro návrháře a obsahuje popis chování uvnitř systému. Pro model chování se používají DFD, ERD a DD, vytváří se hierarchická struktura DDF, která se následně vyvažuje. V posledním kroku se doplní o STD diagramy a minispecifikaci a dokončení datového slovníku.

Po dokončení modelu chování může být vytvořen implementační model. Zde se určí, které procesy budou automatizovány a které budou manuální (v modelu vystupují jako terminátory a v reálu reprezentují uživatele, který procesy manuálně provádí).

SSADM – Structured System Analysis and Design Method

Vyvinuta firmou LBMS, velmi rozšířená ve Velké Británii, kde se stala standardem. Vývoj software je rozdělen do šesti následujících etap:

- analýza stávajícího systému,
- specifikace požadavků,
- výběr technických možností,
- návrh logických dat,
- návrh logických procesů,
- fyzický návrh.

Hlavními nástroji pro modelování systému jsou diagramy datových toků, logické datové struktury (LDS) a životní cykly entit (ELH). Pro DFD používá odlišnou notaci.

Data Structured System Development – DSSD

Datově orientovaný přístup k analýze. Nejlepších výsledků je dosaženo, když struktura programu odpovídá hierarchické struktuře datového modelu. K datovému modelování používá diagramy entit (obdoba ERD).

Programy pro podporu strukturální analýzy

Microsoft Visio

Dia

<http://live.gnome.org/Dia>

CASE Studio Toad Modeler

<http://www.casestudio.com/enu/default.aspx>

Visual Paradigm

<http://www.visual-paradigm.com/>

Edge Diagrammer

<http://www.pacestar.com/>

Literatura

1. Tietze, P.: strukturální analýza – úvod do projektu řízení. Grada, Praha 1992. ISBN: 80-854-24-45-2
2. Straka, M.: Vývoj databázových aplikací. Grada, Praha 1992. ISBN: 80-85424-43-6.
3. Mikulášek, A.: Modely strukturované analýzy pro nevidomé. Bakalářská práce, Masarykova univerzita, Fakulta informatiky, Brno 2008. Dostupné z http://is.muni.cz/th/134645/fi_b/Bakalarska_prace.txt
4. Řepa, V.: Podnikové procesy - procesní řízení a modelování. Grada Publishing a.s.