Obsah

[Informační systémy, jejich základní vlastnosti a typy 2](#_Toc357789827)

[Informační systémy 2](#_Toc357789828)

[Vlastnosti (které by kvalitní IS s maximální výkonností měl splňovat) 3](#_Toc357789829)

[Typy informačních systémů 3](#_Toc357789830)

[Analýza informačních systémů (IS), role modelování a metodik při tvorbě IS 4](#_Toc357789831)

[Role modelování a metodik při tvorbě IS 4](#_Toc357789832)

[Metodika návrhu a realizace informačního systému – strukturální a objektová analýza 6](#_Toc357789833)

[Strukturální analýza 6](#_Toc357789834)

[Objektová analýza 7](#_Toc357789835)

[Typy přístupů v návrhu IS 8](#_Toc357789836)

[Metodologie 8](#_Toc357789837)

[Datové modelování, perzistence objektů, konceptuální a fyzický datový model 9](#_Toc357789838)

[Datové modelování 9](#_Toc357789839)

[Perzistence objektů 9](#_Toc357789840)

[Konceptuální datový model 9](#_Toc357789841)

[Fyzický datový model 9](#_Toc357789842)

[Datové sklady (Data Warehousing), OLAP systémy, jejich význam a oblasti využití, základními principy, dimenze, agregace, extrakce a transformace dat, srovnání transakčních a analytických systémů (OLAP a OLTP technologií) 10](#_Toc357789843)

[Datové sklady 10](#_Toc357789844)

[Charakteristika 10](#_Toc357789845)

[Databázový systém – OLTP (Online Transaction Processing) 11](#_Toc357789846)

[DataWarehouse – OLAP (Online Analytical Processing) 11](#_Toc357789847)

[Využití DW 11](#_Toc357789848)

[Architektura DataWarehouse 12](#_Toc357789849)

[Datová tržiště (Data Mart) 12](#_Toc357789850)

[Systém OLAP (OnLine Analytical Processing) 12](#_Toc357789851)

[MOLAP - Multidimenzionální OLAP 13](#_Toc357789852)

[ROLAP – Relační OLAP 13](#_Toc357789853)

[Vlastnosti CASE nástrojů a jejich význam v analýze a návrhu informačních systémů 14](#_Toc357789854)

[CASE nástroje 14](#_Toc357789855)

[Dělení CASE systémů 14](#_Toc357789856)

[Dělení CASE systémů dle rozsahu možností 14](#_Toc357789857)

[Vlastnosti CASE systémů 15](#_Toc357789858)

[Použití CASE systémů 15](#_Toc357789859)

[Některé příklady CASE systémů 15](#_Toc357789860)

# Informační systémy, jejich základní vlastnosti a typy

## Informační systémy

Informační systém (IS) je systém pro sběr, udržování, zpracování a poskytování informací.

Informační systémy jsou založené na informačních a komunikačních technologiích. V poslední době se používá zkratka ICT (Information and Communication Technologies).

IS obsahuje data, znalosti a informace:

* **Data**
	+ jakékoli vyjádření (reprezentace) skutečnosti
	+ schopné přenosu, uchování, interpretace či zpracování
	+ umožňují přenášet a zpracovávat odraz skutečnosti
* **Znalosti**
	+ výsledek poznávacího procesu
	+ předpoklad uvědomělé činnosti
	+ to, co jednotlivec ví po osvojení dat a po jejich začlenění do souvislostí
	+ účel znalostí - porozumět realitě
* **Informace**
	+ je definovaná pomocí dat a znalostí
	+ data, která mají smysl/význam
	+ znalosti, které jsou sdělitelné (komunikovatelné)

**Funkce informačního systému**

Konkrétní procesy (činnosti) podporující základní cíle informačního systému:

* získávání informací
* zpracování informací (evidence, organizace – pořádání, kategorizace, konverze –změna média, třídění, vyhledávání, agregace, odvozování nových informací, dolování znalostí)
* uložení informací (zaznamenávání a archivace dat, datová úložiště a datové sklady)
* přenos informací (v rámci počítačových sítí)
* zpřístupnění informací (tisk, zobrazení, vizualizace, šíření…)

## Vlastnosti (které by kvalitní IS s maximální výkonností měl splňovat)

* Musí obsahovat nutné informace, které uchovává, analyzuje a s potřebnou rychlostí předává procesům
	+ informace se týkají zejména vlastní činnosti firmy
	+ např.: výroba, evidence zákazníků, zásob, zaměstnanců, finance, stav a vývoj výrobků
* Musí obsahovat informace o konkurenci, světovém trhu, trendech výroby, optimalizaci výrobních procesů, o místech působnosti firmy, o strategických cílech a podobně.
* Musí obsahovat moduly pro zjednodušení a urychlení výroby
	+ urychlení a zefektivnění návrhu výrobků, technologická příprava výroby a její řízení
* Musí umožňovat rychlou komunikaci pracovníků firmy, jednotlivých pracovních úseků, ale musí také zahrnovat komunikaci se světem.
* Musí umožňovat z dostupných informací zpracovávat cíle a strategie firmy, koordinovat činnost různých procesů a tím přispívat k zefektivnění činnosti firmy.
* Musí nabízet rychlou komunikaci se zákazníkem přes počítačovou síť.
* Musí obsahovat další nutné moduly k vedení firmy, jako jsou statistiky, mzdy, účetnictví, kompletní personalistika, sklad, oblast manažer – marketing, výroba a další.

## Typy informačních systémů

**Rozdělení dle funkce**

* Systémy ERP (Enterprise Resource Planning)
* Systémy na podporu rozhodování
	+ např. BI (Business inteligence)
* Systémy na podporu plánování
	+ např. SCM (Supply Chain Management)
* Systémy řízení vztahů se zákazníky
	+ např. CRM (Customer Relationship Management
* Systémy pro tvorbu a správu dokumentů
	+ např. PDM (Product Data Management), PLM (Product Lifecycle Management)
* Systémy na podporu návrhu a projekční činnosti
	+ např. CAD systémy
* CASE Systémy (Computer Aided Software Engineering - Počítačová podpora SI)
* Knihovní systémy
	+ např. eLibrary (rozpovídat se o dokumentografických databázích)

**Rozdělení dle technologie zpracování dat**

* OLTP - umožňují skupině uživatelů vykonávat bezprostředně (online) velké množství transakcí
* relační databáze
* OLAP - analýza velkého množství údajů, většinou jen pro čtení, nadstavba OLTP

**Rozdělení dle uživatelů**

* Veřejné informační systémy
	+ Informační systémy, které jsou dostupné široké veřejnosti a poskytují veřejné informační služby
	+ V tomto smyslu se jedná o jakékoli informační systémy bez ohledu na jejich provozovatele, obsah, typ, formu a příp. cenu poskytovaných informací a služeb
* Privátní, uzavřené, neveřejné informační systémy
	+ např. podnikové informační systémy, systémy zajišťující obranu státu

# Analýza informačních systémů (IS), role modelování a metodik při tvorbě IS

## Role modelování a metodik při tvorbě IS

Složitost systému se promítá do složitosti jeho návrhu a realizace. (tedy i do modelování)

Role modelování a metodiky je taková, že odstraňuje tyto problémy (asi)

* Systém dělá něco jiného než by měl
* Systém řeší problémy lokálně
	+ důsledkem je, že jedna a tatáž věc je řešena na různých místech několikrát, po každé jinak
* Opravy a změny systému jsou velmi obtížné a drahé. (Jestliže opravujeme chybu na základě lokálních znalostí, tak vlastně opravujeme výskyt chyby, ale ne její příčinu.)
* Systém nelze realizovat několika skupinami současně, paralelně. Bez plánu vznikají komunikační problémy.

**Metodiky**

Chceme-li se vyhnout potížím s lokálním rozhodováním, musíme **postupovat metodicky** (ne chaoticky), strukturálně, dle „dobrých“ osvědčených vzorů.

Návod jak postupovat nám dávají **ověřené postupy** – vypracované metodiky.

Metodiky odrážejí určité náhledy na „realitu“, říkají „**jaké**“ kroky učinit v jakém pořadí a „**jak**“ je provádět. Dobré metodiky nám říkají i „**proč**“ to tak má být.

Metodiky jsou **konservovanou zkušeností** několika generací programátorů a projektantů. Zobecnění principů, zásad, které se osvědčily, viz historie UML.

**Modely**

Místo abychom se snažili popsat systém jako celek, vytváříme na něj **jednotlivé pohledy** – jeho jednotlivé, dílčí modely. Díváme se na systém postupně z jednotlivých „míst pozorování“, z jednotlivých perspektiv.

Díváme-li se na systém z jednoho místa, opomíjíme vlastnosti z tohoto místa „neviditelné“, nepodstatné a tím si práci zjednodušíme tak, že je mentálně zvládnutelná. Jednotlivé pohledy jsou jednodušší, zvládnutelné. Opomíjené vlastnosti se neztratí, jsou hlavními vlastnostmi v jiných pohledech - modelech.

Pohledy musíme volit tak, že postupně popíšeme všechny relevantní vlastnosti systému. Postupně popíšeme vše, co potřebujeme k dosažení stanoveného cíle.

Z jednotlivých pohledů lze zpětně zrekonstruovat celý systém (počítačová tomografie).

Pro tvorbu různých pohledů jsou obvykle **využity diagramy** – grafické objekty, jejichž kombinací lze tyto pohledy vytvářet.

**Diagram je graficky znázorněný model.** Diagram popisuje jistou část modelu pomocí grafických symbolů.

Tento přístup lze přirovnat k modelu stavby, který je tvořen syntézou dílčích stavebních plánů odpovídajících specifickým pohledům na stavbu – plán hrubé stavby, plánu rozvodů elektřiny, plánu rozvodů vody, … V každém z těchto plánů jsou zobrazeny pouze elementy modelu podstatné pro daný pohled, od ostatních elementů modelu je abstrahováno. **Pohledy nejsou nezávislé**, dohromady tvoří konzistentní pohled na systém, tedy konzistentní model.

Pro tvorbu diagramů systému, jejichž syntézou bude model, definuje např. **UML devět typů diagramů**.

# Metodika návrhu a realizace informačního systému – strukturální a objektová analýza

Existuje v zásadě několik metodik, které popisují analýzu, návrh a realizaci informačních systémů. Jedná se o více méně podrobný popis postupu, který vede návrháře jasně definovanými fázemi krok po kroku při vytváření IS. Metodiky jsou často obecné a každá firma si je může přizpůsobit pro vlastní potřebu a prostředí. Mezi nejpoužívanější patří Strukturální analýza (ta je nejstarší), SSADM (vznik 80. léta ve Velké Británii) a Objektová analýza (konec 80. let, 90. léta)

## Strukturální analýza

Postup se zaměřuje na data a jejich transformaci pomocí procesů systému (hlavními nástroji jsou tedy DFD (Data Flow Diagram) popisující procesy a toky dat a ERD (Entity Relationship Diagram) popisující data a vztahy mezi nimi). Nejznámější je Yourdonova strukturální analýza. Ta se zaměřuje na vytvoření logického modelu nového navrhovaného systému (esenciální model) a následně přizpůsobení implementačním požadavkům. Nedoporučuje vycházet z fyzického modelu nahrazovaného systému (např. oproti SSADM).

* Proces vytváření modelu současného systému může vyžadovat tolik času a úsilí, že uživatel se stane velmi frustrovaným a netrpělivým, což může vést až ke zrušení projektu.
* Někteří uživatelé, manažeři a programový analytici považují systémovou analýzu za plýtvání času před pravou prací (vytvářením zdrojového kódu).
* Někteří uživatelé pochybují o smyslu analýzy systému, který má být nahrazen novým.
* Někteří analytici se nechají unést modelováním a snaží se vytvořit naprosto detailní model současného systému, což vede k velkému plýtvání časem.

**Esenciální model**

Kompletní logický model (bez implementačních detailů) navrhovaného systému

Skládá se z:

* Model prostředí (Environmental model)
	+ Definuje hranice systému a okolí
	+ Obsahuje kontextový diagram a seznam událostí (event list)
* Model chování
	+ Definuje vnitřní chování systému, tak aby plnil požadavky okolí
	+ Používané modely (diagramy) DFD, ERD, DD (datový slovník), SP (specifikace procesů), případně STD (stavový diagram)

**Funkční přístup**

Chápe smysl modelu reálného světa v tom, že obsahuje souhrn stavů reálného světa a změn těchto stavů. Funkční přístup je dynamický. Ke změnám stavů v modelu dochází prostřednictvím operací (to je abstraktní obraz událostí). Operace jsou podle různých hledisek a principu sdružovány do vyšších celků - funkcí. Příklad: Data Flow Diagram (DFD, Diagram datových toků).

**Datový přístup**

Zaměřuje na vlastnosti (atributy) reálného světa, jejichž abstraktním obrazem v IS jsou data. Datový přístup je statický.

Příklad: Entity-relationship model (ERM). Datovým modelem reálného světa je potom systém entit, charakterizovaných jejich atributy a jejich vzájemnými vztahy. Smyslem modelování z hlediska datového přístupu je především formulovat ideální (konceptuální) podobu uspořádání dat v informačním systému, která je „věrným“ obrazem reálného světa.

## Objektová analýza

Zaměřuje se na objekty reálného světa a jejich třídy a komunikaci mezi objekty. V 90. letech navrženo během 5 let až 40 OO metodik. Metody měli tolik společného, že se tři klíčoví autoři (Booch, Jacobson a Rumbaugh) rozhodli své návrhy integrovat do jedné metodiky **RUP (Rational Unified Process)**. Vychází ze specifikace požadavků, snažíme se v průběhu analýzy najít všechny objekty relevantní pro navrhovaný systém a vztahy mezi nimi. Ze začátku jde pouze o hrubý nástin objektů (doménový model), v pozdějších fázích se třídy zpřesňují až na úroveň objektů reprezentujících fyzické třídy programovacího jazyka.

**Nástroje objektové analýzy**

* Často se používá modelovací jazyk UML
	+ Model případů užití
	+ Doménový model
	+ Diagram tříd
	+ Stavové diagramy, sekvenční diagramy a další
* CRC karty (Class-Responsibility-Collaboration cards)

Umožňuje zvládnout návrh i složitých a velkých systémů

* Hierarchie objektů – sdružovány podle logických souvislostí do balíků a podbalíků
* Přirozený přechod od analýzy k návrhu

Pro vytvoření objektového návrhu je zapotřebí:

* Identifikace objektů a tříd
	+ Založeno často na analýze textového popisu řešeného problému
		- podstatná jména jsou kandidáti na objekty
		- slovesa na metody objektů a interakci objektů
* Příklady objektů – faktura, zaměstnanec, zakázka, atd.
* Nalezení chování (metod) objektů
* Nalezení vazeb mezi objekty a jejich uspořádání do hierarchií
	+ Při pohledu na objekty získané shora uvedenými metodami je možno rozpoznat u některých společné rysy, opakující se metody, atd.
	+ Tyto charakteristiky se v objektovém modelu převedou do hierarchie vazeb dědičnosti a/nebo celek-část (reprezentace charakteristik objektů), podle pravidel správného objektového návrhu.

**Doménová analýza**

* Hledají se objekty, operace a vazby, které znalci z problémové oblasti (např. uživatelé nebo správci stávajících systémů, příslušní úředníci, ...) pokládají za důležité (často používají jejich názvy apod.).
* Konstrukce obecného modelu domény (tj. např. systému výplaty mezd) podle konzultací se znalci (bude obsahovat jak používané objekty a jejich vztahy, tak standardní vzory chování).
* Zkoumání stávajících systémů pro doménu -> inspirace. Identifikace podobností a odlišností mezi nimi podle konzultací.

## Typy přístupů v návrhu IS

**Zdrojový (datový)**

* Stanovení datového modelu reality a nového systému (zajímají nás uživatelské potřeby) -> informační (datová) analýza

**Funkční přístup**

* Stanovení cílů, které by měl nový systém splňovat (zajímá nás fungování systému) -> funkční analýza

**Top-down**

* Postup od abstraktního ke konkrétnímu

**Bottom-up**

* Postup od konkrétního k obecnému

**Strukturovaný**

* Aplikace má podobu hierarchie funkcí, která je realizována strukturovanými programy. Styl práce AKCE -> OBJEKT

**Objektově orientovaný**

* Funkce jsou realizovány pomocí "programových fragmentů" nad stabilními strukturami databází. Styl práce OBJEKT -> AKCE

**Rigorózní (striktní)**

**Agilní (light)**

**Plánovací**

**Konstrukční**

**Implementační**

## Metodologie

**Strukturovaná metodologie**

* Entitně relační model (ERA, ERD, E-R)

Grafický prostředek pro analýzu a zobrazení datového modelu systému

* Diagram datových toků (DFD - Data Flow Diagram)

Grafický prostředek návrhu a zobrazení funkčního modelu systému

**Objektová metodologie**

Objekt - cokoliv reálného či abstraktního s jasně vymezenou rolí v daném kontextu. Objekt má 3 charakteristiky:

* Stav
* Identita
* Chování

Vlastnosti objektů:

* Sebeidentifikace
* Zapouzdření (atributů a funkcí do objektu)
* Klasifikace (objekty jsou sdruženy do tříd)
* Dědičnost
* Polymorfismus (volání přetížených metod, metod se stejným jménem u různých tříd pomocí rozhraní)

Přínosy:

* Podpora složitých objektů
* Možnost opakovaného použití objektů

# Datové modelování, perzistence objektů, konceptuální a fyzický datový model

## Datové modelování

* jedna ze základních funkcí IS je ukládání a následné zpracování informací ve formě dat, proto se provádí při návrhu IS také datové modelování
* jeho úkolem je zvolit jaké objekty, jako nositele informace, potřebujeme ukládat do trvalé paměti a jaké jejich vlastnosti a vztahy mezi nimi chceme uchovávat
* při datovém modelování obvykle vytváříme konceptuální a fyzický datový model a také určujeme způsob perzistence objektů

## Perzistence objektů

* zabývá se, kam a jakým způsobem budou objekty trvale uloženy
* objekty se obvykle ukládají do relační databáze ve formě datových záznamů v tabulkách
* pro programový přístup do databáze se často používá standardizované softwarové API pro databáze jako ODBC nebo JDBC
* pro usnadnění programátorské práce při vytváření perzistentní vrstvy můžeme využít objektově-relační mapování, které nám zajistí automatickou transformaci ukládaných objektů do záznamů relační databáze - např. framework Hibernate pro Java aplikace

## Konceptuální datový model

* vyjadřuje jaké objekty a jejich atributy budeme ukládat a vztahy mezi nimi
* má vyjádřit esenci – podstatu systému
* říká, co musí systém dělat, aby zajistil uživatelovy požadavky
* implementačně nezávislý
* využití: údaje potřebné pro specifikaci zadání úlohy nebo pro komunikaci s uživatelem
* pro grafické vyjádření se používají ERA modely (diagramy)

## Fyzický datový model

* odvozuje se z konceptuálního modelu a vyjadřuje navíc, jak přesně budou data v konkrétní databázi uložena
* využití: údaje potřebné pro projektanta při algoritmizaci a programování
* také se popisuje ERA modely, které obsahují i přesné databázové typy atributů tabulek

# Datové sklady (Data Warehousing), OLAP systémy, jejich význam a oblasti využití, základními principy, dimenze, agregace, extrakce a transformace dat, srovnání transakčních a analytických systémů (OLAP a OLTP technologií)

Základní problémy u běžných transakčních databázových systémů:

* nedosažitelnost dat skrytých v transakčních systémech
* dlouhá odezva při plnění komplikovaných dotazů
* složitá, uživatelsky nepříjemná rozhraní k databázovému softwaru
* cena v administrativě a složitost v podpoře vzdálených uživatelů
* soutěžení o počítačové zdroje mezi transakčními systémy a systémy podporujícími rozhodování
* Cesta k řešení těchto problémů = datové sklady, tzv. Data Warehouse – DW

## Datové sklady

* samostatný informační systém postaven na již pořízených datech, určen především k jejich analýze
* architektura založená na relačním SŘBD, která se používá pro údržbu historických dat získaných z databází operativních dat, jenž byla sjednocena a zkontrolována před jejich použitím v databázi DW
* data z DW jsou aktualizována v delších časových intervalech, jsou vyjádřena v jednoduchých uživatelských pojmech a jsou sumarizována pro rychlou analýzu
* DW je obrovská databáze obsahující data za dlouhé časové období
* často slučuje data z více rozdílných zdrojů, které mohou obsahovat data různé kvality nebo používat nejednotné formáty a reprezentace
* objemově zabírá stovky GB až několik TB
* nemusí být databází v běžném smyslu, tj. pro přesné provádění transakcí
* je určen pro rychlé vyhledávání
* nejsou kladeny nijak důrazné požadavky na správnost a úplnost dat

## Charakteristika

* data jsou uložena na různých místech ve formě relačních tabulek
	+ uživatelé mohou tabulky jen číst
	+ zapisovat může aktualizační program pravidelně udržující tabulky
* dotazy jsou většinou komplexní
	+ podporují tzv. on-line analytické zpracování (OLAP)
		- výrazně se liší od on-line transakčního zpracování (OLTP)
		- operační databáze je přizpůsobena pro podporu OLTP
			* složité OLAP dotazy by vyústily do nepřijatelné odezvy
		- typické OLAP operace
			* roll-up (snížené stupně detailu -> zvýšení stupně agregace)
			* drill-down (zvýšení stupně detailu o informaci)
			* slice-and-dice (selekce a projekce - změna kombinací dimenzí, které jsou zobrazovány)
			* pivot (přeorientování vícerozměrného pohledu na data)



* na základě dotazu se pospojují potřebná data do vícerozměrné tabulky (nebo více tabulek), do kterých lze klást SQL dotazy
* pro častější dotazy si uchovávají předem připravené vícerozměrné tabulky
* zátěž je většinou způsobena složitými dotazy, jež přistupují k miliónům záznamů a provádějí množství operací
* data bývají modelována vícerozměrně
	+ v obchodním data warehouse mohou těmito rozměry být např. čas prodeje, místo prodeje, prodavač, výrobek, …
	+ rozměry mohou být i hierarchické např. čas prodeje jako den-měsíc-čtvrtletí-rok, zboží jako výrobek-kategorie-průmysl
	+ spojení více tabulek pomocí odkazu na řádky jednotlivých tabulek
	+ používají speciální organizaci dat, přístupové a implementační metody, jež obecně nejsou v komerčních databázových systémech určených pro OLTP podporovány

## Databázový systém – OLTP (Online Transaction Processing)

* zákaznicky orientovaný
* aktuální data -- lze považovat i za slabinu, při výpadku (chybě), vznikají ztráty pro byznys
* ER schéma
* sofistikované atomické transakce i přes několik systémů(bank, po síti,...)
* velikost DB až několik GB
* jednoduché a efektivní
* příkladem je bankomat

## DataWarehouse – OLAP (Online Analytical Processing)

* orientovaný na trh, rychlé (oproti OLTP) získání výsledků na analytické dotazy
* historická data, multidimenzionální datový model
* agregovaná data (nenormalizovaná=redundantní)
* schéma hvězdy či vločky
* převážně pouze čtení
* velikost až TB
* použití: byznys reporty o prodeji, marketing, management reporty, rozpočty, finanční předpovědi a reporty

## Využití DW

* prezentace dat
* testování hypotéz
* objevování nových informací

## Architektura DataWarehouse

* tři úrovně:
1. klient
2. OLAP server (MOLAP/ROLAP server)
3. databázový server DW
* data lze organizovat v tzv. multidimenzionálním datovém modelu
	+ odlišný od modelu relačního
	+ odpovídá mu specializovaný software, multidimenzionální SŘBD (MDD)
	+ model připomíná techniku spreadsheet ve více než dvou rozměrech
	+ data jsou implementována pomocí vícerozměrných polí, jejichž dimenze odpovídají dimenzím podnikání organizace
* navržení a vytvoření DW je proces skládající se z následujících bodů:
1. definovat architekturu, umístění a rozčlenění dat a fyzickou organizaci
2. naplánovat kapacitu, vybrat OLAP servery a nástroje
3. spojit servery, klientské nástroje, zdroje přes gatewaye, drivery ODBC, ...
4. navrhnout schéma a pohledy, přístupové metody, některé složité dotazy
5. mít skripty pro získávání, čištění, transformaci, ukládání a aktualizaci dat
6. vytvořit koncové uživatelské aplikace
7. spustit data warehouse i aplikace
* vytvoření je složitý proces trvající mnohdy i několik let
* mnoho organizací proto používá Data Mart umožňující rychlejší práci



## Datová tržiště (Data Mart)

* část datového skladu sloužící konkrétnímu účelu nebo určená konkrétní organizační jednotce
* přístupová vrstva data warehousu
* DW slouží jako základna pro extrakci množin dat, resp. jejich agregaci do dílčích (replikovaných) MDD (Multidimenzionální DB)
	+ MDD může pro DW sloužit ve dvou rolích
		- ”front-end” pro DW a poskytovat uživateli služby pro realizaci analytického zpracování (DW/OLAP)
		- “front-end” jednomu (několika) systémům OLTP - alternativa za DW, tj. poskytnout uživateli s OLTP data analytickým způsobem (OLTP/OLAP) – jde vlastně o datové tržiště

## Systém OLAP (OnLine Analytical Processing)

* na databázové stroje jsou kladeny specifické požadavky
1. objem zpracovávaných dat
	* transakční systém o velikosti gigabajtů dosáhne použitím jen jedné dimense velikosti desítek či stovek gigabajtů
2. rychlost odezvy analytického systému je důležitá
3. počet uživatelů současně pracujících s databází není zajímavý
	* počet pracovníků vyššího managementu je omezen
	* pro pracovníky nižších stupňů bývají údaje z datových skladů převedeny do menších specializovaných databází – datových tržišť
* s těmito omezeními se vyrovnává dvojím způsobem
1. uzpůsobení stávajících systémů pro práci s vícerozměrovými daty
* přidáním modulu, který to zajišťuje a prostředků pro jeho ovládání
* v lepším případě mění způsob uložení dat, v horším“překládá” operace s vícedimenzionálními daty na operace s daty relačními
1. vytvoření speciálního systému správy dat, určeného pouze pro OLAP
* umožňuje provést maximum optimalizací vzhledem k nárokům, jež jsou kladené analytickým způsobem práce - převažující způsob

**Programy pro vytváření a plnění databáze**

* převodní programy
	+ načtení data z několika databází, či souborů a udělat z nich novou databázi, agregace se musí naprogramovat
* systémy znázorňující převodu dat graficky a administrátor dat namapuje zdrojová data do struktur vytvářeného datového skladu
	+ výsledkem jsou buď programy (scripty) nebo přímo vykonání funkce
* moduly pro plánování jednotlivých akcí

**Nástroje pro práci s daty - poslední trendy v architektuře klient/server**

* nabízejí variantu tenkého klienta v podobě HTML prohlížeče

**Reporting, monitorování, ad-hoc dotazy**

* programy umožňující kladení dotazů a formátování odpovědí
	+ nejčastěji jde o vizuální dotazovací nástroje
	+ makra v tabulkovém procesoru
	+ uživatelské rozhraní různě propracované:
		- zadání seskupení výsledku podle různých kriterií
		- formální kontrola dotazů
		- vytváření slovníků a metadat

## MOLAP - Multidimenzionální OLAP

* datová krychle (obsahuje fakta)
* hierarchické dimenze (částečné či totální uspořádání)
	+ vločkové schéma -- hlavní tabulka faktů je v relaci s dimezionálními tabulkami, přes cizí klíče, dimenzionální tabulky mohou být také v relaci s dalšími subdimenzionálními tabulkami podobně jako hlavní tabulka faktů; vytváří hierarchie dimenzí
	+ hvězdové schéma -- je speciální případ vločkového, dimenzionální tabulky již nejsou v relaci s dalšími subdimenzionálními tabulkami; žádné hierarchie, jednodušší

## ROLAP – Relační OLAP

* na relační architektuře založený model DW strukturou propojených DB tabulek - Relační OLAP (ROLAP) – pomalejší zpracování než MOLAP
* užívá relační nebo rozšířený relační DBMS, např. server METACUBE Informix, pracuje s relačními tabulkami uspořádanými do hvězdy/vločky, adresuje pomocí klíče, data jsou neagregovaná)

# Vlastnosti CASE nástrojů a jejich význam v analýze a návrhu informačních systémů

**CASE** Computer Aided Software (System) Engineering (Počítačová podpora softwarového inženýrství)

Modely softwarových systémů jsou příliš složité

* nutná podpora různých úrovní abstrakce, různých pohledů
* nutnost rozdělení mezi jednotlivé vývojáře

## CASE nástroje

* slouží pro standardizaci používaných postupů
* nástroj na podporu práce analytiků a programátorů při vývoji informačních systémů, zejména ve fázi analýzy a návrhu – tvorba modelů
* mezistupeň mezi analýzou problému a programováním
* označení pro integrovanou tvorbu programových aplikací pomocí programových prostředků s minimální potřebou manuálního psaní zdrojového kódu programu

**obsah**: databáze (repository, systémová encyklopedie) navrhovaných prvků informačního systému

**funkce**: podpora realizace projektu informačního systému

**základ**: metodika = návod na vytváření modelů a určení závislostí mezi nimi

## Dělení CASE systémů

* Nižší CASE – podpora tvorby software
	+ návrhy obrazovek, formulářů, menu
	+ jazyková podpora
* Vyšší CASE – podpora analýzy a návrhu
	+ tvorba diagramů a modelů
	+ kontrola konzistence modelu
	+ Příklad: AxiomSys: strukturovaná analýza
* Integrované CASE – podpora životního cyklu softwaru
	+ od analýzy po generování kódu
	+ round-trip engineering
	+ podpora tvorby dokumentace
	+ Příklad: Oracle Designer
* Komponentové CASE – otevřenost
	+ repository s rozhraním (SCM, testování)
	+ integrace nástrojů od různých výrobců
	+ Příklad: Rational Suite Enterprise

## Dělení CASE systémů dle rozsahu možností

* Jedna fáze
	+ podpora jedné fáze ŽC (analýza, prog.)
* Jedna metodika
	+ podpora dané metodiky přes ŽC
* Více fází, více metodik
	+ transformace modelů, vlastní metodiky
* Vývoj + management
	+ včetně podpůrných funkcí pro řízení

## Vlastnosti CASE systémů

**Kladné**

* Zvýšení produktivity
	+ automatizace prací - čas na podstatné věci
	+ lepší přehled o modelu a implementaci
	+ podpora rozdělení práce
	+ snazší údržba dokumentace i systému
* Zvýšení kvality
	+ podpora analýzy - lepší znalost požadavků
	+ kontroly konzistence a úplnosti
	+ synchronizace reality a dokumentace
	+ podpora používání standardů

**Záporné**

* Cena
	+ CASE jsou drahé (řádově 100.000 Kč)
	+ vybírat s rozvahou (reklamní slogany vs. skutečné možnosti vs. skutečné potřeby)
* Doba návratnosti investice
	+ na počátku potřebné zaškolení a zaučení
	+ přínosy obvykle až od 2-3 projektu
* Změna stylu práce
	+ nástroj bez používání metodiky k ničemu
	+ nutnost podřídit se CASE (techniky, metoda)
	+ podpora managementu nutná
	+ jen kód je aktuální (pomalé updatování modelu při změnách)

## Použití CASE systémů

* automatizovaná evidence vytvořených objektů a specifikací, dokumentace vývoje systému
* grafická podpora modelování (notace)
* kontrola správnosti modelů podle zvolené metodiky, zajištění integrity a konzistence návrhu
	+ předem definovaná integritní omezení a jejich kontrola
	+ automatické uplatnění změn vytvořených v jedné části ve všech souvisejících částech návrhu
* podpora týmové práce
	+ identifikace osob a týmů zodpovědných za jednotlivé modely
	+ tvorba více modelů současně
	+ současná práce více osob na jednom modelu
* správa verzí
* automatický převod definovaných modelů do konkrétního logického a fyzického návrhu
	+ generování programu
	+ popisu databáze
	+ příp. prototypu
* reverse engineering
	+ zpětné generování konceptuálního modelu z existující aplikace

## Některé příklady CASE systémů

* Powerdesigner (Sybase)
* Together (Borland)
* Oracle designer (Oracle)
* Rational Rose (Rational Software Corporation)
* Enterprise Architect (Sparx)