SSZ 2016 SWI

Databázové technologie

DB

Část: informační systémy

Seznam otázek

1. Informační systémy, jejich typy a základní vlastnosti.
2. Analýza informačních systémů (IS), role modelování a metodik při tvorbě IS.
3. Metodiky analýzy a návrhu IS – sběr požadavků, objektová analýza (diagram tříd, diagram případu užití, sekvenční diagram).
4. Datové modelování, perzistence objektů, konceptuální a fyzický datový model.
5. OLAP systémy, jejich význam a oblasti využití, základními principy.
6. Vlastnosti a typy CASE nástrojů a jejich význam v analýze a návrhu informačních systémů.

1. Informační systémy, jejich typy a základní vlastnosti.

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

IS obsahuje data, znalosti a informace:

- Data
  - jakékoli vyjádření (reprezentace) skutečnosti
  - schopné přenosu, uchování, interpretce 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 IS:

- 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 IS měl splňovat)

- Musí obsahovat nutné informace, které uchovává, analyzuje a s potřebnou rychlostí předává procesům (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 zákazník (např. CRM - Customer Relationship Management)
- Systémy pro tvorbu a správu dokument (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 bez prostředně (online) velké množství transakcí
- Relační databáze
- OLAP - analýza velkého množství údajů, většinou jen pro čtení, nadstavba OLTP

Rozdělení dle uživatelů:

- Veřejné informační systémy
    - IS, 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

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

IS viz předchozí otázka

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:

- 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**.

3. Metodiky analýzy a návrhu IS – sběr požadavků, objektová analýza (diagram tříd, diagram případu užití, sekvenční diagram).

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.

Sběr požadavků

- Analýza trhu, marketingové průzkumy
- Rozhovory s uživateli
    - předem připravený rozhovor, který vede moderátor (klade otázky, dává slovo)
    - nedoporučuje se více než 2 hodiny
    - předem si připravit scénář, které okruhy se budou probírat, v jakém pořadí, scénář se snažit nenásilně dodržovat
- Pozorování, práce s uživateli
    - pozorování prací u zákazníka (účast analytiků)
    - analýza existujícího se systému
- Dotazníky
- Studium dokumentace stávajících systémů
- Studium hlášených problémů
- Předvedení prototypu
- Sběr požadavků na základě případů užití (use-case)
    - srozumitelné pro většinu lidí (jak pro analytiky, tak pro zákazníka)
    - součástí notace UML, podpora v CASE nástrojích

Objektová analýza

Zaměřuje se na objekty reálného světa a jejich třídy a komunikaci mezi objekty.

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

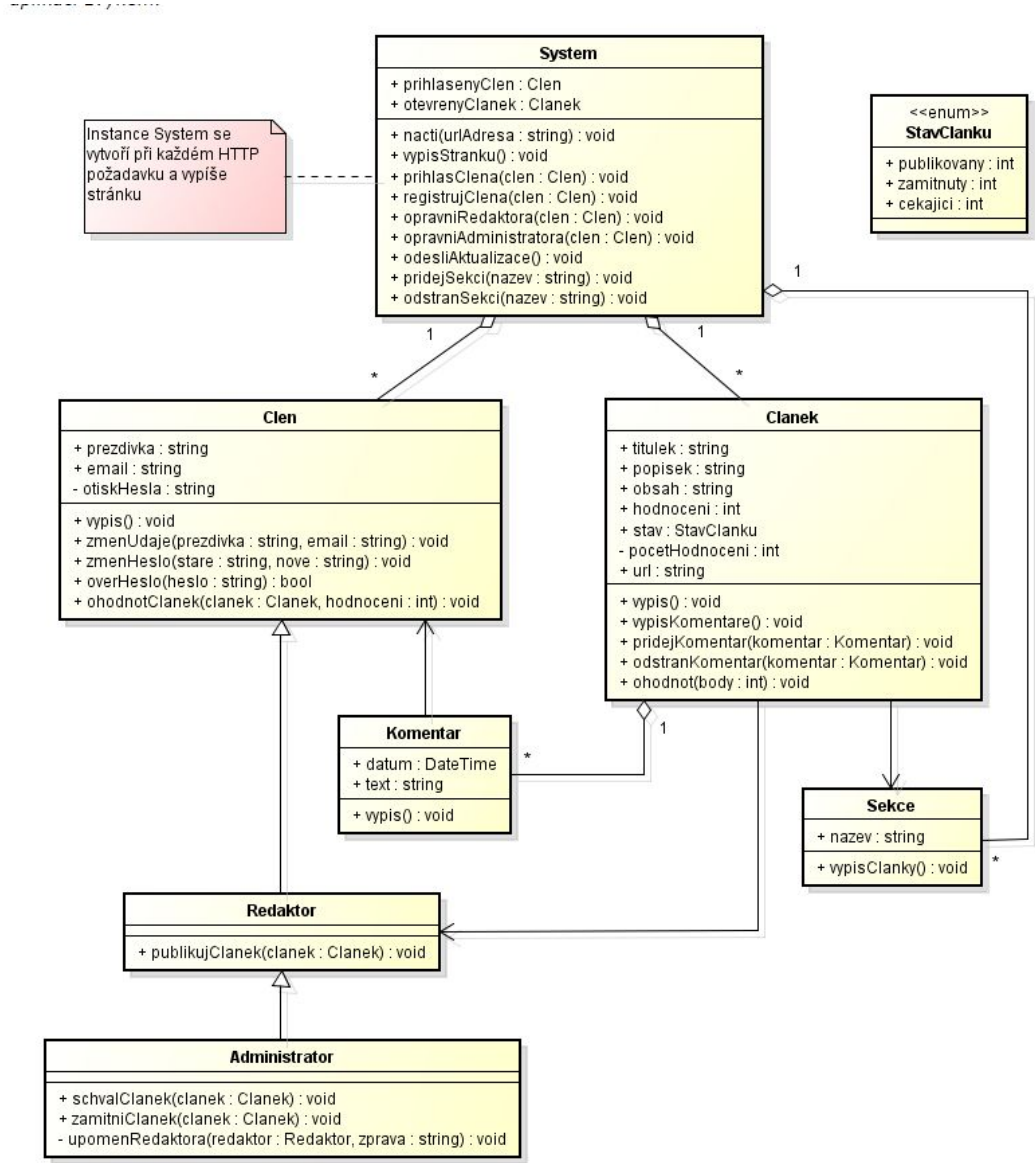
- Hierarchie objektů – sdružovány podle logických souvislostí do balíků a podbalíků
- Přirozený přechod od analýzy k návrhu

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.

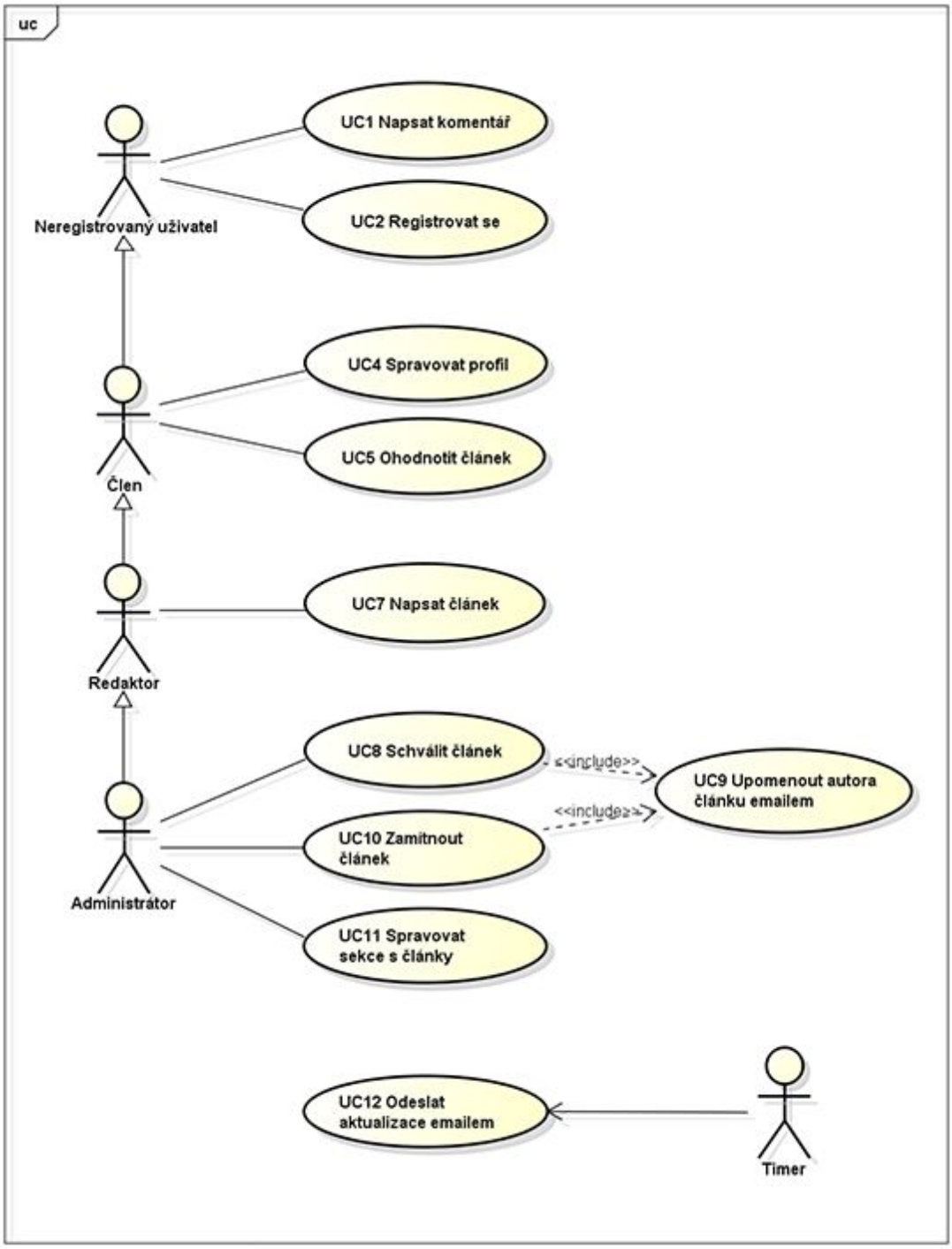
Diagram tříd

Diagram tříd je diagram implementace. To je rozdíl oproti doménovému modelu, který byl spíše náčrt systému. Class diagram je již naostro, musí být úplný, když ho programátor přepíše do kódu, kód musí fungovat. Budeme zde mít tedy všechny třídy, které program bude obsahovat. Třídy budou mít všechny atributy a také metody. Diagram je **platformově závislý**, tedy specifický pro určitý programovací jazyk. Mimo jiné to znamená, že se v identifikátorech již nevyskytuje diakritika, atributy mají datové typy specifické pro daný jazyk a podobně.



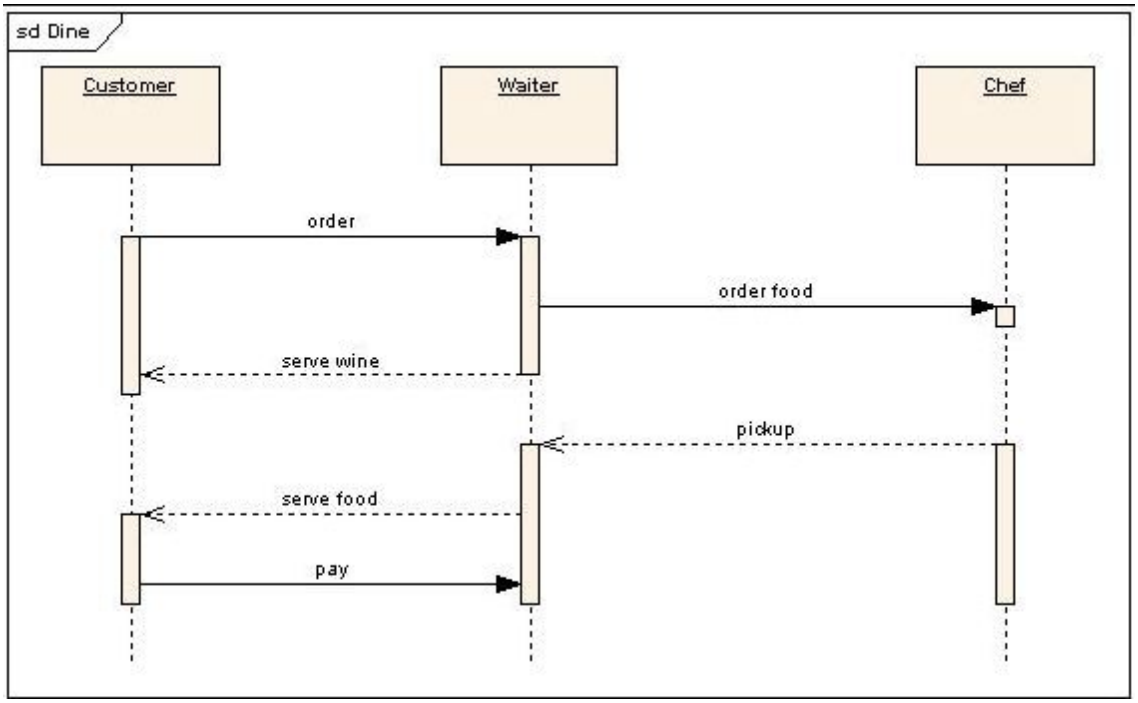
Případ užití

Use Case Diagram (česky diagram případů užití) zobrazuje **chování systému** tak, **jak ho vidí uživatel**. **Účelem diagramu** je **popsat funkcionalitu systému**, tedy co od něj klient nebo my očekáváme. Diagram vypovídá o tom,**co má systém umět**, ale **neříká jak to bude dělat**.



Sekvenční diagram

Sekvenční diagram je jeden z diagramů interakcí jazyka UML. Zachycuje časově uspořádanou posloupnost zasílání zpráv mezi objekty. Sekvenční diagram nejčastěji znázorňuje spolupráci několika vzorových objektů v rámci jednoho případu užití.



4. 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

5. OLAP systémy, jejich význam a oblasti využití, základními principy.

Systém OLAP (OnLine Analytical Processing)

Je technologie uložení dat v databázi, která umožňuje uspořádat velké objemy dat tak, aby byla data přístupná a srozumitelná uživatelům zabývajícím se analýzou obchodních trendů a výsledků (Business Intelligence). Způsob uložení dat se svým zaměřením liší od běžněji užívaného **OLTP** (**Online Transaction Processing**), kde je důraz kladen především na snadné a bezpečné ukládání změn v datech v konkurenčním (více-uživatelském) prostředí.

Na databázové stroje jsou kladeny specifické požadavky

- objem zpracovávaných dat
  - transakční systém o velikosti gigabajtů dosáhne použitím jen jedné dimense velikosti desítek či stovek gigabajtů
- rychlost odezvy analytického systému je důležitá
- 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

- uzpůsobení stávajících systémů pro práci s více-rozměrovými 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
- 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

- 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)

6. Vlastnosti a typy 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

Typy CASE nástrojů

- 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 životního cyklu (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 systém

- 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)