

SKRIPTA

INFORMAČNÍ SYSTÉMY

Mgr. Jan Matula PhD.



2015

Obsah

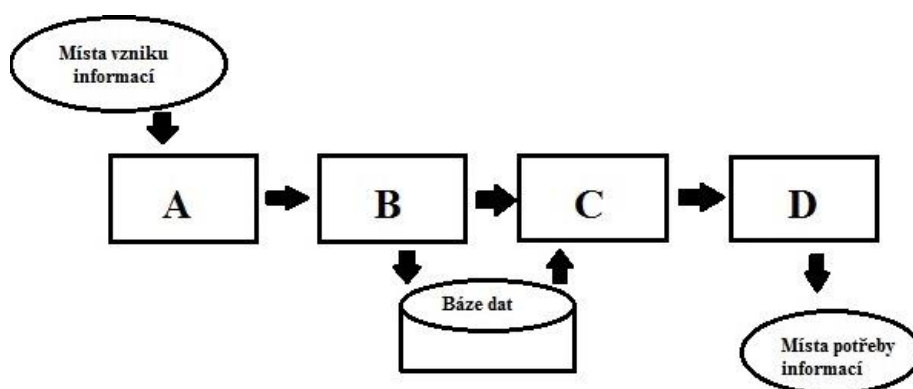
| | |
|---|----|
| 1. <i>INFORMAČNÍ SYSTÉMY</i> | 3 |
| 2. <i>ÚVOD DO TEORIE SYSTÉMŮ</i> | 11 |
| 3. <i>DATABÁZOVÉ MODEL Y</i> | 16 |
| 4. <i>METODIKA – VÝVOJOVÁ KLASIFIKACE</i> | 25 |
| 5. <i>STRUKTUROVANÝ, OBJEKTIVĚ ORIENTO VANÝ PŘÍSTUP</i> | 32 |
| 6. <i>PODNIKOVÝ INFORMAČNÍ SYSTÉM</i> | 39 |
| 7. <i>TRENDY VE VÝVOJI ERP</i> | 44 |
| 8. <i>TESTOVÉ OTÁZKY</i> | 49 |

1. INFORMAČNÍ SYSTÉMY

Obecná charakteristika IS

- systém umožňující komunikaci a transformaci informací - časové, prostorově i co do formy tak, aby byly lépe využity než v původním stavu (systém, který přidává hodnotu k zpracovávaným či komunikovaným informacím)
- speciální typ komunikačního média, jehož cílem je odstranit bariéry v přístupu k informacím
- účelové uspořádání vztahu a informačních toků mezi informačními zdroji, lidmi a technologickými prostředky spolu s procesy zpracování a komunikace informací
- model reálného světa, jehož základními prvky jsou informace

Konceptuální model IS



Jaké informace zpracovává IS?

- **strukturované data popisující neprostorové objekty** (záznamy v databázích, souborech apod.) - dělíme na numerické a nenumerické
- **strukturované data popisující prostorové objekty ve formě souřadnic** (geografické informační systémy) - převážně numerické data
- **nestrukturované data** (volné texty, záznamy rozhovorů apod.)
- **metadata** (popis dat pomocí SGML jazyků – HTML, XML, struktury typu MARC, Dublin Core apod.), které jsou často spojené s nestrukturovanými daty (plné texty dokumentů typu články, zpráva, kniha...) nebo obrázky, mapami, schémata, multimediálními dokumenty atd.

Typické problémy řešené IS

- potřeba informací (pro poznání, pro rozhodování, pro realizaci určité činnosti)
- složitost (complexity)
- znovu použitelnost (reusability)
- automatizace
- komunikace
- bezpečnost, spolehlivost, minimalizace rizik...

Automatizovaný IS

- informační systém fungující s podporou informačních a komunikačních technologií
- automatizace procesu
- digitalizace datové základny

Úvod do problematiky

- s rozvojem lidského poznání roste množství informací
- pro efektivní práci s informacemi začaly vznikat specializované IS
- IS = systémy pro sběr, uchování, vyhledávání a zpracování informací (dat, údajů) za účelem jejich poskytování

Historie IS – agendové zpracování

- Rozvoj spjat s vývojem ICT zejména PC
- Na počátku – zpracování velkých informačních objemů na jednom PC = SYSTÉMY HROMADNÉHO ZPRACOVÁNÍ DAT nebo AGENDOVÉ ZPRACOVÁNÍ.
- Data se zaznamenávala do formulářů, následoval přepis na vhodné médium (děrné štítky, disketa), poté primární a sekundární zpracování, výsledkem jsou tištěné výstupní sestavy.

Agendové zpracování dat

a) ZÁVISLOST DAT A PROGRAMŮ

- Každý program řeší nejen vlastní aplikační problém, ale i formát **fyzického uložení** dat na médiu.
- **Navazující úlohy** musí **respektovat již vytvořené – deklarované fyzické struktury dat**.
- Při změně datové struktury v jednom programu je nutné měnit a kompilovat i všechny další programy, které s touto strukturou pracují, i když se v jejich funkčnosti nic nemění.
- Nízká efektivnost datových struktur i programu.

b) PROBLÉMY AGENDOVÉHO ZPRACOVÁNÍ

- **Redundance:** některé informace ve více souborech opakují, jsou redundandní. Redundance je zdrojem mnoha dalších problémů
- **Konzistence:** vzájemná shoda údajů. Postupem času – vlivem nedostatečné kontroly v programech se stejné hodnoty na různých místech v datových souborech, začnou rozcházet.
- **Integrita:** data aktuální, odrážejí skutečnost z reálného světa. Problémem tedy je **zabezpečit, aby chybou či nedůsledností uživatele** nebyla porušena integrita a konzistence dat.
- **Obtížná dosažitelnost dat:** aplikační programy pro konkrétní požadavky; pro nový požadavek nutno napsat nový aplikační program - bez pomoci programátora nelze.

- **Izolovanost dat** - data roztroušena v různých souborech, soubory mohou být různě organizovány, data různě formátována. To komplikuje tvorbu nových aplikačních programů a možnost realizovat vazby mezi datovými strukturami.
- **Současný přístup více uživatelů:** větší systémy vyžadují současný přístup k datům více uživatelů. Pak je nutné, aby programy vzájemně spolupracovaly, jejich činnosti byly koordinovány.
- **Ochrana proti zneužití:** při zpracování důvěrných či tajných dat není přípustné, aby měl kdokoliv přístup ke všem informacím. Při klasickém zpracování však musí mít programátor aplikačních programů k dispozici tolik podrobností, že to ochranu dat prakticky znemožňuje.

c) POUČENÍ Z AGENDOVÉHO ZPRACOVÁNÍ

- Vedlo k návrhu a vytvoření programových systémů (systému řízení báze dat – SŘBD) následujících vlastností:
 1. existuje **seznam datových typů**, které jsou v programovém systému definovány; pro tyto typy dat programový systém **vytváří fyzickou strukturu** na disku a automaticky řeší všechny přístupy k datům
 2. existují prostředky pro definování všech **sledovaných vlastností** popisovaných objektu
 3. existuje **soubor instrukcí**, které nad definovanými daty provádějí jednotlivé operace; každá instrukce je vlastně mohutnou procedurou, v níž je řešen fyzický přístup k datům i realizace vlastní operace; jinak než prostřednictvím systému není možno s daty pracovat
 4. programový systém řeší způsob, jak zaznamenat **vztahy mezi objekty**

Úvod do terminologie

Systém řízení báze dat (SŘBD) - lze chápat jako souhrn procedur a datových struktur, které zajišťují nezávislost databázových aplikací na detailech vytváření, výběru, uchování, modifikaci a zabezpečení ochrany databází na fyzických paměťových strukturách počítače.

Pro práci s daty SŘBD podporují zejména tyto funkce:

- vytvoření báze dat (CREATE)
- vkládání dat (INSERT)
- aktualizace dat (UPDATE)
- rušení dat (DELETE)
- výběr z báze dat (SELECT)

Dále je podporována tvorba **formulářů** (vstupních obrazovek, Forms), **výstupních sestav** (Reports) a aplikačních programů.

Data

údaje, které mají určitou vypovídací schopnost. Mohou být určitým způsobem uspořádány (seřazeny, např. podle velikosti, chronologicky atd.) a jsou uživateli k dispozici v různých formách (tabulky, grafy, zvukové signály, grafická forma atd.). Data jsou obvykle rozdělena na dílčí údaje (**atributy**) o dané množině objektů (entit), na základě nichž lze získat určitou informaci, která může vést k rozhodovacímu procesu.

Záznam

je souhrn údajů (**atributů**) o dané části objektu, které jsou uloženy **v položkách (polích, Fields)** charakterizovaných názvem a datovým typem. Význam pojmu **záznam (Record)** a **položka** lze snadno ukázat na bázi dat STUDENT.

Abychom mohli u každého studenta zaznamenat potřebné údaje, musí mít záznam 5 položek, které mohou mít tyto názvy:

- identifikační číslo
- jméno
- příjmení
- datum narození
- bydliště
- pohlaví

Datové typy

každá položka musí být určitého datového typu. Obecně akceptované jsou tyto typy dat:

- **Textový typ** - textový řetězec, zpravidla do max. délky 255 znaků
- **Číselné typy** - pro uložení celých a reálných čísel s pevnou i plovoucí desetinnou tečkou
- **Logický typ** - slouží k uložení logické hodnoty Ano/Ne (True/False, Yes/No)
- **Memo** - pro uložení textu proměnné délky
- **Datumový typ** - pro uložení datumových a časových hodnot.

Model

- je souhrn pravidel pro reprezentaci logické organizace dat v databázi
- Základní typy modelů: hierarchický, síťový, relační, objektový

KONCEPTUÁLNÍ MODEL je formalizovaný popis zájmové reality. Popisuje fakta o reálném světě, která jsou v čase neměnná nebo se mění pouze málo. Nejedná se o popis dat v počítači. Důvodem zavedení konceptuálního schématu je odstranit náhlý přechod od zájmové reality přímo k logickému schématu báze dat.

RELAČNÍ MODEL – rozšíření

- byl popsán v roce 1970 Dr. Coddem. V současnosti je tento model **nejčastěji využíván u komerčních SŘBD**. Relační databázový model má jednoduchou strukturu. Data jsou

organizována v tabulkách, které se skládají z řádků a sloupců. Všechny databázové operace jsou prováděny na těchto tabulkách.

Dr. Codd definoval jako minimalisticky relační ty systémy, které splňují tyto dvě vlastnosti:

1. Databáze je chápána uživatelem jako množina relací a nic jiného
2. V relačním SŘBD jsou k dispozici minimálně operace selekce, projekce a spojení, aniž by se vyžadovaly explicitně předdefinované přístupové cesty pro realizaci těchto operací

12 pravidel pro relační SŘBD

Dále definoval Dr. Codd dalších 12 pravidel pro relační SŘBD:

- 1) **Informační pravidlo** - všechny informace v relační databázi jsou vyjádřeny explicitně na logické úrovni jediným způsobem - hodnotami v tabulkách.
- 2) **Pravidlo jistoty** - všechna data v relační databázi jsou zaručeně přístupná kombinací jména tabulky s hodnotami primárního klíče a jménem sloupce.
- 3) **Systematické zpracování nulových hodnot** - nulové hodnoty jsou plně podporovány relačním SŘBD pro reprezentaci informace, která není definována a to nezávisle na datovém typu.
- 4) **Dynamický on-line katalog založený na relačním modelu** - popis databáze je vyjádřen na logické úrovni stejným způsobem jako zákaznická data, takže autorizovaný uživatel může aplikovat stejný relační jazyk ke svému dotazu jako uživatel při práci s daty.
- 5) **Obsáhlý datový podjazyk** - relační systém může podporovat několik jazyků a různých módů použitých při provozu terminálu. Nicméně musí být nejméně jeden příkazový jazyk s dobře definovanou syntaxí, který obsáhle podporuje definici dat, definici pohledů, manipulaci s daty jak interaktivně, tak programem, integritní omezení, autorizovaný přístup k databázi, transakční příkazy apod.
- 6) **Pravidlo vytvoření pohledů** - všechny pohledy, které jsou teoreticky možné, jsou také systémem vytvořitelné.
- 7) **Schopnost vkládání, vytvoření a mazání** - schopnost zachování relačních pravidel u základních i odvozených relací je zachována nejen při pohledu na data, ale i při operacích průniku, přidání a mazání dat.
- 8) **Fyzická datová nezávislost** - aplikační programy jsou nezávislé na fyzické datové struktuře.
- 9) **Logická datová nezávislost** - aplikační programy jsou nezávislé na změnách v logické struktuře databázového souboru.
- 10) **Integritní nezávislost** - integritní omezení se musí dát definovat prostředky relační databáze nebo jejím jazykem a musí být schopna uložení v katalogu a nikoliv v aplikačním programu.

11) Nezávislost distribuce - relační SŘBD musí být schopny implementace na jiných počítačových architekturách.

12) Pravidlo přístupu do databáze - jestliže má relační systém jazyk nízké úrovně, pak tato úroveň nemůže být použita k vytváření integritních omezení a je nutno vyjádřit se v relačním jazyce vyšší úrovně.

ARCHITEKTURA DATABÁZÍ

U databází rozlišujeme 3 základní architektury:

- a) **Centrální architektura**
- b) **Architektura file – server**
- c) **Architektura klient - server**

a) Centrální architektura

- data i SŘBD jsou uložena v centrálním počítači.

Tato architektura je typická pro terminálovou síť, kdy se po síti přenáší vstupní údaje z terminálu na centrální počítač do příslušné aplikace, výstupy z této aplikace se přenáší na terminál. Protože aplikační program i vlastní zpracování probíhá na centrálním počítači, který může zpracovávat více úloh, mají odezvy na dotazy určité zpoždění.

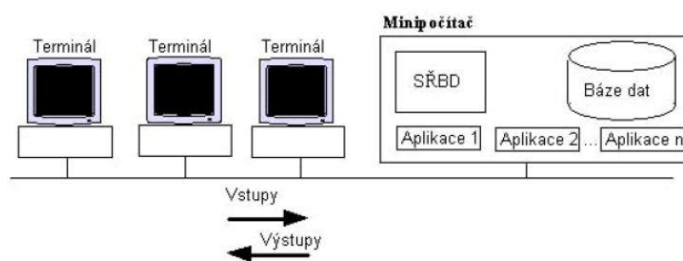


Schéma centrální architektury

b) Architektura file - server

Tato metoda souvisí zejména s rozšířením osobních počítačů a sítí LAN. SŘBD a příslušné databázové aplikace jsou provozovány na jednotlivých počítačích, data jsou umístěna na file serveru a mohou být sdílena. Aby nedocházelo ke kolizím při přístupu více uživatelů k jednomu datům, musí SŘBD používat vhodný systém zamykání (položek nebo celých tabulek).

Komunikace uživatele se systémem probíhá následujícím způsobem:

1. uživatel zadá dotaz
2. SŘBD přijme dotaz, zasílá požadavky na data file-serveru,
3. file-server posílá bloky dat na lokální počítač, kde jsou data zpracovávána podle zadaného dotazu (vyhledávání, seřídění atd.),
4. výsledek dotazu se zobrazí na obrazovce osobního počítače.

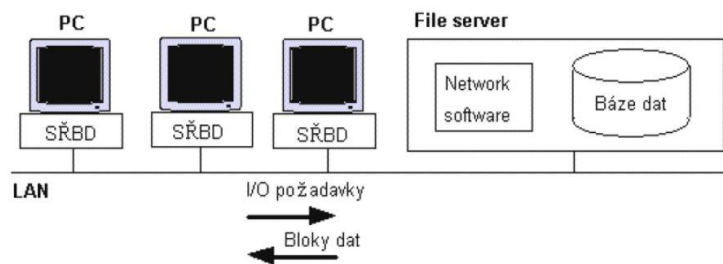


Schéma architektury file – server

c) Architektura klient – server

Architektura klient - server je v podstatě založena na lokální síti (LAN), personálních počítačích a databázovém serveru. Na personálních počítačích běží program podporující např. vstup dat, formulaci dotazu atd. Dotaz se dále předává pomocí jazyka SQL (Structured Query Language) na databázový server, který jej vykoná a vrátí výsledky zpět na personální počítač. Databázový server je tedy nejvíce zatíženým prvkem systému a musí být tvořen dostatečně výkonným počítačem.

V podstatě je založena na lokální síti (LAN), personálních počítačích a databázovém serveru. Na personálních počítačích běží program podporující např. vstup dat, formulaci dotazu atd.

Dotaz se dále předává pomocí jazyka SQL (Structured Query Language) na databázový server, který jej vykoná a vrátí výsledky zpět na personální počítač. Databázový server je tedy nejvíce zatíženým prvkem systému a musí být tvořen dostatečně výkonným počítačem.

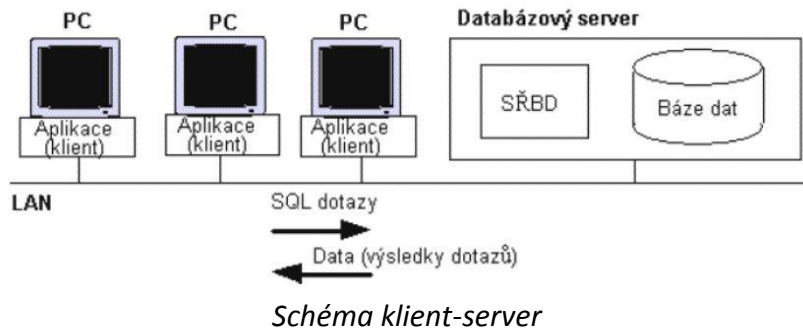
Komunikace probíhá tímto způsobem:

1. uživatel zadává dotaz (buď přímo v SQL, nebo musí být do tohoto jazyka přeložen)
2. dotaz je odeslán na databázový server
3. databázový server vykoná dotaz
4. výsledek dotazu je poslán zpět na vysílací počítač, kde je zobrazen

Architektura klient-server redukuje přenos dat po síti, protože dotazy jsou prováděny přímo na databázovém serveru a na personální počítač jsou posílány pouze výsledky.

Např.: pokud je mezi 10 000 záznamy pouze 100 záznamů, které splňují podmínku dotazu, pak na personální počítač putuje pouze těchto 100 záznamů. V případě architektury file-server je však nutné poslat všech 10 000 záznamů na personální počítač, tam se teprve provede dotaz a zpracuje nalezených 100 záznamů.

Architektura klient-server vyhovuje i náročným aplikacím a je využívána většinou renomovaných databázových firem.



DISTRIBUOVANÁ DATABÁZE

Distribuovaná databáze je množina databází, která je uložena na několika počítačích. Uživatelé se však jeví jako jedna velká databáze.

Distribuovanou databázi charakterizujeme 3 vlastnostmi:

- Transparentnost
- Autonomnost
- Nezávislost na počítačové síti

DD - TRANSPARENTOST

Z pohledu klienta se zdá, že všechna data jsou zpracovávána na jednom serveru v lokální databázi. Uživatel používá syntakticky shodné příkazy pro lokální i vzdálená data, nespécifikuje místo uložení dat, o to se stará distribuovaný SŘBD.

DD – AUTONOMNOST

- S každou lokální bází dat zapojenou do distribuované databáze je možno pracovat nezávisle na ostatních databázích.
- Lokální databáze je funkčně samostatná, propojení do jiné části distribuované databáze se v případě potřeby zřizují dynamicky.
- V distribuované databázi neexistuje žádný centrální uzel nebo proces odpovědný za vrcholové řízení funkcí celého systému, což výrazně zvyšuje odolnost systému proti výpadkům jeho částí.

DD – NEZÁVISLOST NA POČ. SÍTI

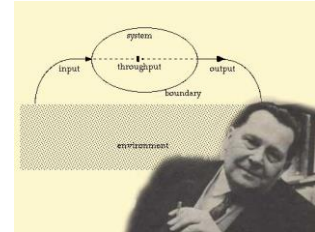
Jsou podporovány různé typy architektur lokálních i globálních počítačových sítí (LAN, WAN). V jedné distribuované databázi tedy mohou být zapojeny počítače i počítačové sítě různých architektur, pro komunikaci se používá jazyk SQL.

2. ÚVOD DO TEORIE SYSTÉMŮ (od přírodních věd k vědě informační)

Systémový přístup

- Ve 20. stol. mechanistický přístup nahrazen systémovým přístupem.
- Hlavní představitel:

Ludwig von Bertalanffy (1901–1972),
rakouský biolog, *General System Theory*



- Systémový přístup zaměřený na vztahy, vazby, vzájemné vlivy mezi částmi celku.
- (Inspirace *strukturalismem*, vztah mezi formou a obsahem.)

Systém a struktura

Systém – ohraničena množina prvků a vazeb mezi nimi, které společně určují vlastnosti celku.

„Wikidefinice“: „*Systém (česky soustava) je souhrn souvisejících prvků, sdružený do nějakého smysluplného celku. Systém se skládá z částí, které jsou spojeny za účelem umožnění toku informací, materiálu nebo energie.*“

Struktura – vztahy mezi jednotlivými prvky systému.

Změna struktury mění vlastnosti systému.

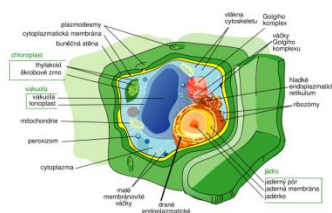
A co *emergence*? Odkud se vynořuje? Kolik neuronů myslí?

Systém

Systém vs. objekt/jsoucno

- na objektu lze pozorovat množství (sub)systémů (člověk – nervový systém, imunitní systém, buňka jako systém...);
- sám prvek systému je považován za černou schránku, podstatné jsou pouze vazby s okolím;
- systémový přístup zkoumá existující svět v celé složitosti a komplexnosti (je univerzálním nástrojem poznávání?)

Co je to za systém?



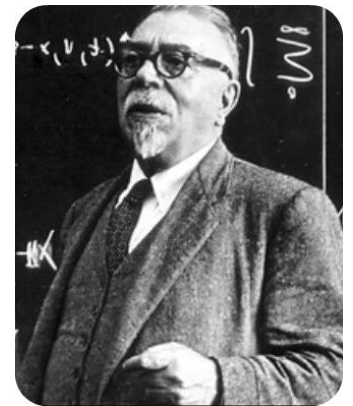
System a zpětná vazba

(Kybernetika vs. Biologie)

Zpětná vazba je termín pro situaci, kdy výstup nějakého systému ovlivňuje zpětně jeho vstup.

Norbert Wiener (1894–1964) přirovnával zpětnovazební smyčku ke slepecké holi, která dává slepci zpětnou informaci o jeho pohybu a ovlivňuje tak jeho pohyb následující. (*Cybernetics or the Control and Communication in the Animal and the Machine*, 1948)

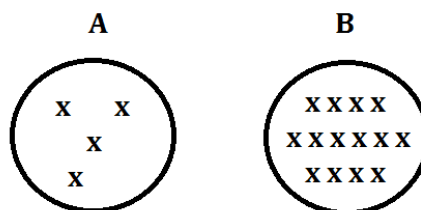
Jak komunikují živé systémy a jak systémy technické?



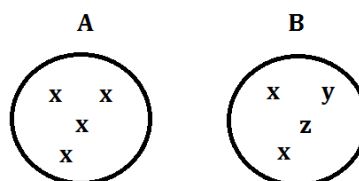
Složitosti systémů

Diferenciace komplexity:

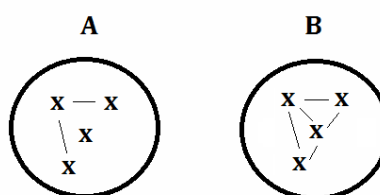
1.) počet prvků



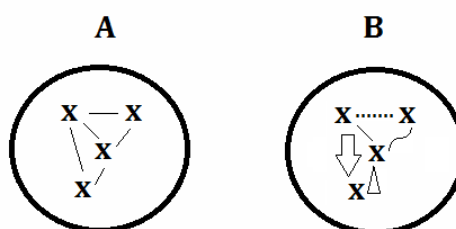
2.) různorodost prvků



3.) počet vazeb



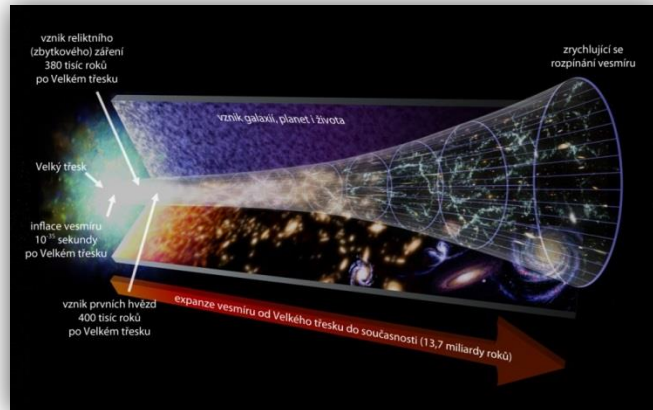
4.) různorodost vazeb



Typy systémů:

- Izolované – absolutně netečné ve vztahu k prostředí;
- Uzavřené – energetická výměna (něco za něco?);
- Otevřené – „relační promiskuita“.

Izolovaný systém?

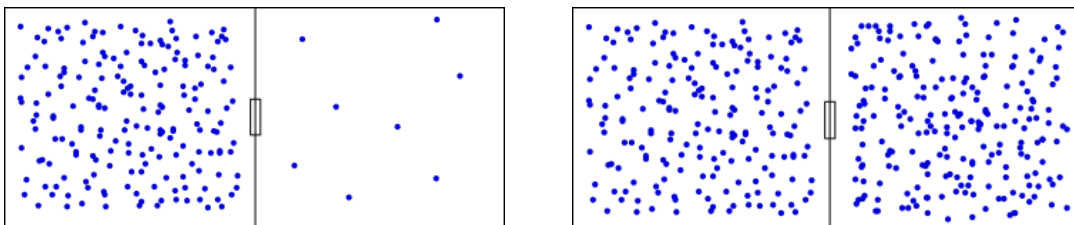


Systémy a evoluce („prokletí“ entropie!)

A jak to všechno souvisí s informací?

(Aneb horor termodynamických zákonů a směr „→ času“!)

Uff, ta prokletá *entropie*!



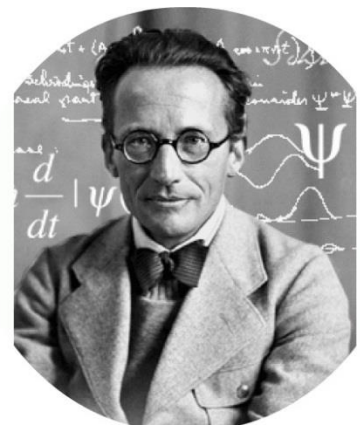
Negentropie pana Schrödingera

Živé (otevřené) systémy odolávají – po určitou dobu – 2. termodynamickému zákonu. Ale jenom relativně.

Vítězství je vždy jen dočasné!

Co je život? (1944)

„Život se zdá být uspořádaným a zákonitým projevem hmoty, jenž se nezakládá výlučně na její tendenci směřovat od pořádku k chaosu, nýbrž dílem na existujícím pořádku, který je udržován.“



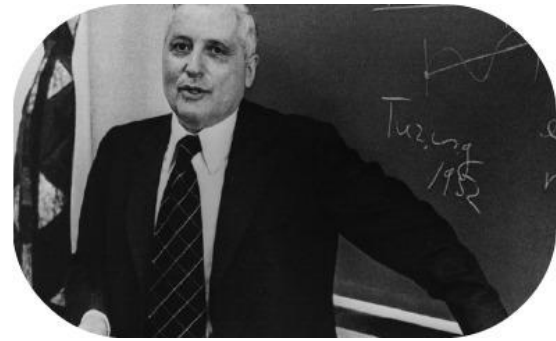
Systemy a chaos (nelineární disipativní struktury)!

Determinismus vs. Chaos – umíme předpovědět vývoj systému? Jak daleko jsme od termodynamické rovnováhy? Uff, bifurkace!

„Neexistuje jednoznačný průběh vývoje.“

„Čím je systém složitější, tím četnější jsou typy fluktuací ohrožujících jeho stabilitu.“

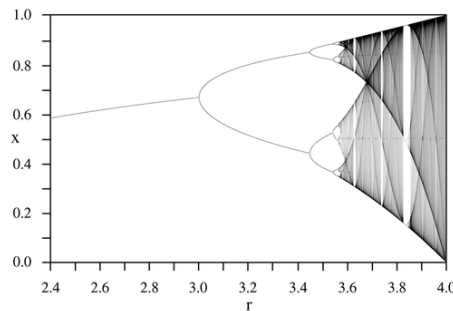
„Růst nevratně probíhající v čase vytváří nevratný vývoj.“



(Ilya Prigogine (1917–2003))

Vývoj komplexních systémů

„**Bifurkace** – označení pro bod zvratu na vývojové linii, kdy v důsledku nerovnováhy negativních a pozitivních zpětných vazeb dojde k rozdělení trajektorie vývoje původní kvality v několik nových struktur, které se kvalitativně liší.“ (zase ta wiki!)



Deterministický chaos

Teorie chaosu – zabývá se chováním jistých nelineárních dynamických systémů, které (za jistých podmínek) vykazují jev známý jako **deterministický chaos** (efekt motýlích křídel).

Chaos jako forma řádu nebo řád jako forma chaosu?

Systém je citlivý na počáteční podmínky – chování se jeví jako náhodné, i když model systému je deterministický v tom smyslu, že je dobře definovaný a neobsahuje žádné náhodné parametry.

Hmm, takže počasí (na dlouho dopředu) nepředpovíme...



Otevřené systémy: autopoietické systémy

Teorie Santiaga: Humberto Maturana (1928) & Francisco Varela (1946–2001)

Řecký pojem *poiésis* – tvorba.

„Všechny živé systémy poznávají.“ „Život je poznání.“

„Živé systémy jsou poznávací systémy a život jakožto proces je proces poznávání. To platí pro všechny organismy, ať mají či



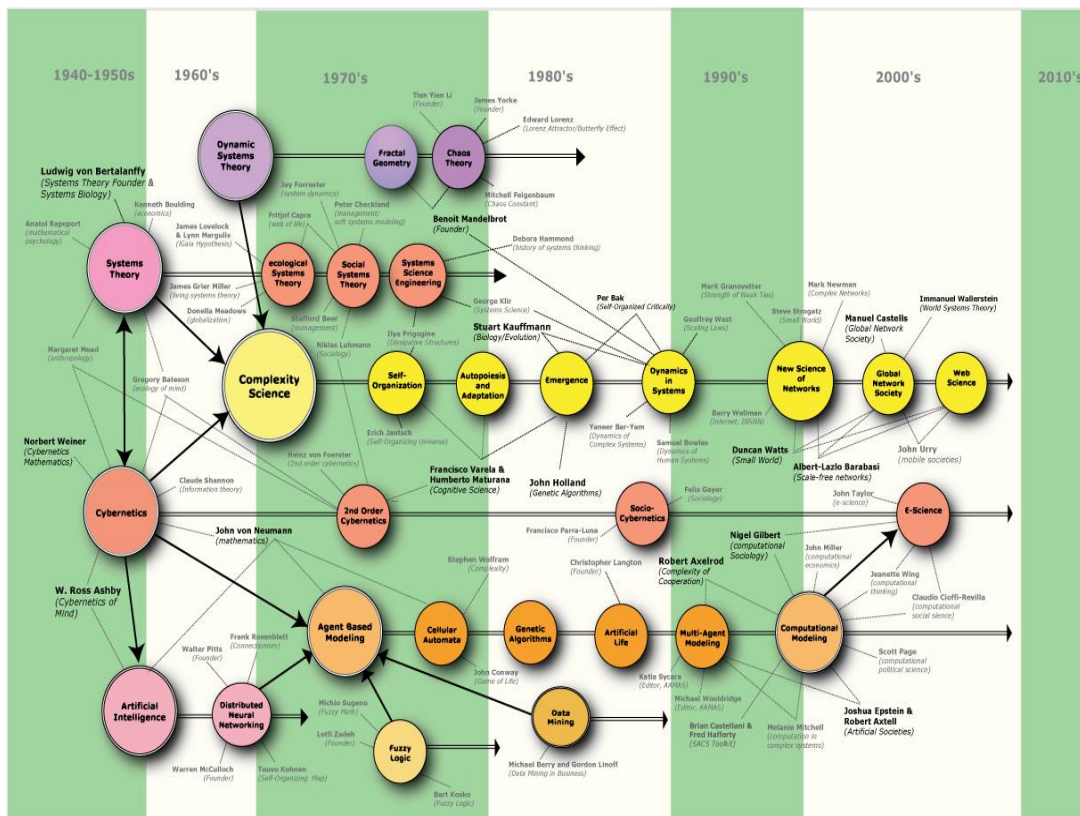
nemají nervovou soustavu.“ (H. Maturana)

„Fyzikální systém lze popsat jako živý, je-li schopen transformovat energii/hmotu do interního procesu autoúdržby a autoreprodukce.“ (F. Varela)

Život je sebeutvářejícím se informačním procesem.

Rizika systémového přístupu!

- systém modeluje skutečnost → model je zjednodušením;
- nebezpečí záměny metody za podstatu předmětu → základ nepřipustné personifikace počítačů, záměny lidské a umělé inteligence apod.
- systém chápaný jako cíl, podstata → neosobnost, odlidštění. Infověda – pojem IS jen metoda, prostředek;
- přirozený svět – nevypočitatelný a nejistý → člověk se v systému může cítit bezpečněji, jistěji → iluze!



3. DATABÁZOVÉ MODELY

(Základní pojmy DTB zpracování)

ENTITA – OBJEKT

- celá posloupnost položek popisuje objekt. Taková struktura položek, která má ucelený význam (zachycuje všechny potřebné údaje o sledovaném objektu) se nazývá **záznamem** (větou, recordem). Je to obvykle skupinová položka.

MNOŽINA ENTIT – MNOŽINA OBJEKTŮ – DATOVÝ SOUBOR – OBSAH TABULKY

- množinu záznamů stejného typu, zaznamenávající ucelenou informaci o množině sledovaných objektů a uloženou na paměťovém médiu, nazýváme **datovým souborem**. Množiny záznamů si můžeme snadno představit ve tvaru **tabulky**, kde každý objekt je popsán jedním řádkem a každý atribut objektu je v jednom sloupci.

DATABÁZE

- množinu datových souborů, uchovávajících data o nějakém uceleném úseku reality, nazýváme **databází**

SYSTÉM ŘÍZENÍ BÁZE DAT – SŘBD

- programový systém (prázdný, bez datových souborů a bez aplikačního programu), umožňující definování datových struktur a datových souborů, řešící fyzické uložení dat ve vnější paměti počítače, umožňující manipulaci s daty a formátování vstupních i výstupních informací, nazýváme **systemem řízení báze dat**.

APLIKAČNÍ ÚLOHA

- **Aplikační úlohou** nad SŘBD nazýváme konkrétní program napsaný pomocí programových prostředků použitého SŘBD nad konkrétní databází, pro tuto úlohu vytvořenou.

INFORMAČNÍ SYSTÉM

- Aplikační úlohy nad společnou databází tvoří ucelený systém, nazývaný **databázovým nebo informačním systémem** (dále jen IS) nad použitým SŘBD.
- V tomto pojetí tedy IS rozumíme celek, řešící rozsáhlejší oblast aplikační, naprogramovaný v jednom SŘBD s vhodně navrženými datovými strukturami tak, aby všechny aplikační úlohy k nim měly optimální přístup. Řeší **uložení, uchování, zpracování a vyhledávání informací a umožňuje jejich formátování** do uživatelsky přívětivého tvaru.

Vztahy mezi entitami (relace)

1:1

V relaci 1:1 odpovídá jednomu záznamu v první tabulce maximálně jeden záznam v druhé tabulce a naopak jednomu záznamu v druhé tabulce maximálně jeden záznam v první tabulce.

| Jméno čtenáře | Číslo čtenáře |
|---------------|---------------|
| Bohumil Novák | 155 |
| Pavel Pokorný | 168 |
| Petr Žáček | 174 |
| Jana Sekyrová | 135 |

| Evidenční číslo | Název knihy |
|-----------------|-------------|
| 1234 | Návrat Orků |
| 1235 | Zapomenutí |
| 1236 | Prvorození |
| 1237 | Nový měsíc |

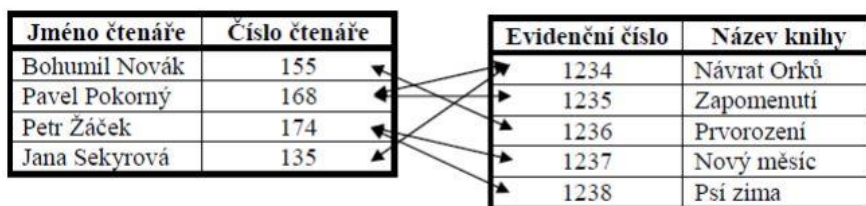
1:N

V relaci 1:N odpovídá jednomu záznamu v první tabulce žádný, jeden nebo více záznamů v druhé tabulce a naopak jednomu záznamu v druhé tabulce maximálně jeden záznam v první tabulce.

N:N

V relaci N:N odpovídá jednomu záznamu v první tabulce žádný, jeden nebo více záznamů v druhé tabulce a naopak jednomu záznamu v druhé tabulce žádný, jeden nebo více záznamů v první tabulce.

Chcete-li vyjádřit relaci typu N:N, musíte vytvořit třetí tabulku, která se často nazývá spojená tabulka, jež rozdělí relaci typu N:N na dvě relace typu 1:N. Primární klíč z těchto dvou tabulek vložíte do třetí tabulky. Výsledkem je, že třetí tabulka zaznamená každý výskyt nebo instanci relace.



Úrovně DBS

Způsoby pohledu na data v databázi:

- FYZICKÝ (INTERNÍ)** – způsob fyzického uložení dat na disk
- KONCEPTUÁLNÍ** – popis struktury databáze (tabulek), popis vztahů mezi uloženými daty
- EXTERNÍ** – popis dat z pohledu uživatele, tj. uživatelské prostředí (formuláře, sestavy,...)

Datové modely

Způsob uložení dat v databázi:

- Hierarchický
- Síťový
- Relační
- Objektový

a) Hierarchický DM

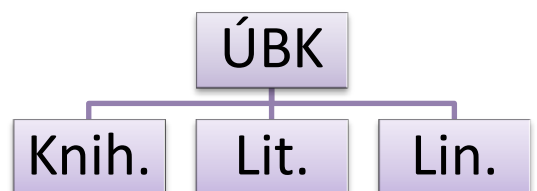
- záznamy jsou organizovány ve stromové struktuře

VÝHODY:

- řeší snadno a rychle vztahy 1:N
- nezáleží na fyzické struktuře dat

NEVÝHODY:

- problémy při řešení vztahů M:N
- problémy při změně struktury dat



b) Síťový DM

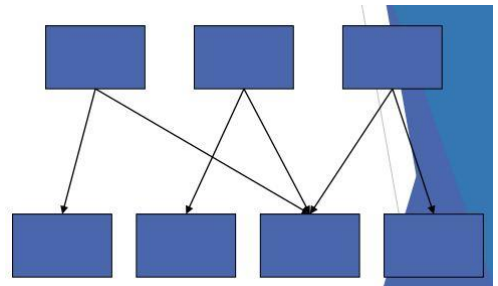
- data jsou reprezentována kolekcemi záznamů a vztahů mezi nimi

VÝHODY:

- řeší snadno a rychle vztahy 1:N i M:N
- nezáleží na fyzické struktuře dat
- rychlé vyhledávání

NEVÝHODY:

- problémy při změně struktury dat



c) Relační DM

RDBS – relační databázový systém

- informace uchovávané v jednom typu objektu jsou uchovávané v tabulkách s určitou strukturou
- tabulky jsou navzájem provázány relacemi
- relace usnadňují vyhledávání různých informací uložených v těchto tabulkách

V SOUČASNOSTI NEJPOUŽÍVANĚJŠÍ MODEL

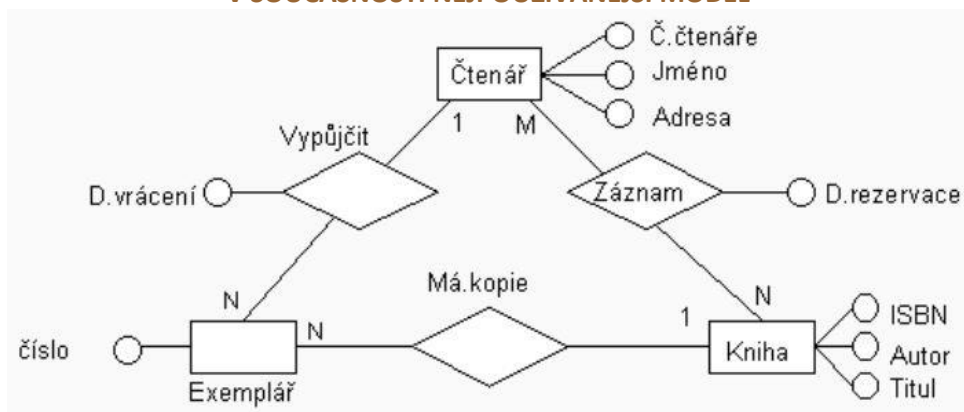


Schéma entitně-relačního modelu

Relační DM

| Číslo objednávky | Název výrobku |
|------------------|---------------|
| 10456 | 003 |
| 10456 | 004 |
| 10457 | 101 |
| 10457 | 002 |
| 10457 | 001 |
| 10457 | 102 |
| 10458 | 006 |
| 10459 | 001 |
| 10459 | 004 |

| Číslo objednávky | Datum objednávky | Id zaměstnance |
|------------------|------------------|----------------|
| 10456 | 12.8.2005 | 01 |
| 10457 | 12.8.2005 | 02 |
| 10458 | 13.8.2005 | 02 |
| 10459 | 13.8.2005 | 03 |

| Id zaměstnance | Jména zaměstnanců | Zaměstnán od |
|----------------|-------------------|--------------|
| 01 | Jan Novák | 1.2.2005 |
| 02 | Emil Král | 15.2.2001 |
| 03 | Václav Nový | 24.8.2002 |

| Id výrobku | Název výrobku | Cena výrobku |
|------------|------------------|--------------|
| 001 | Fausto | 155 |
| 002 | Funghi | 125 |
| 003 | Carpaccio | 135 |
| 004 | Hawai | 140 |
| 005 | Rustica | 139 |
| 006 | Trapolla | 155 |
| 101 | Instalate Beluco | 125 |
| 102 | Instalate Caesar | 128 |

Relační tabulka

atributy

| Id výrobku | Název výrobku | Cena výrobku |
|------------|------------------|--------------|
| 001 | Fausto | 155 |
| 002 | Funghi | 125 |
| 003 | Carpaccio | 135 |
| 004 | Hawai | 140 |
| 005 | Rustica | 139 |
| 006 | Trapolla | 155 |
| 101 | Instalate Beluco | 125 |
| 102 | Instalate Caesar | 128 |

záznamy

d) Objektový DM

- vychází z principů objektově orientovaného přístupu
- Objekt** – datová struktura definovaná jako třída s určitými vlastnostmi a metodami
- komunikace mezi objekty probíhá pomocí zpráv

VÝHODY:

- nejen statické, ale i dynamické chování objektů
- možné vytváření složitějších objektů
- snadnější zadávání dotazů

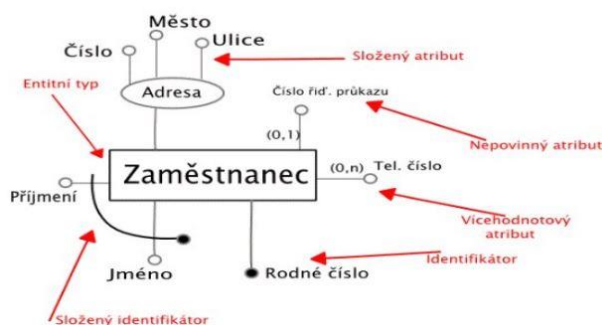
Konceptuální model

- slouží k popisu dat v databázi nezávisle na jejich fyzickém uložení
- umožňuje zobrazit a popsat objekty v databázi a vztahy mezi nimi z hlediska jejich významu a chování
- výsledkem je implementačně nezávislé schéma obecně aplikovatelné v jakémkoli prostředí
- znázorňuje se v podobě **ER diagramu**, který definuje entity (třídy prvků), jejich atributy a relace (vztahy) mezi nimi


Pojmy v ER-modelu

ENTITNÍ TYP (v diagramu) - reprezentuje třídu entit (např. zaměstnanec)

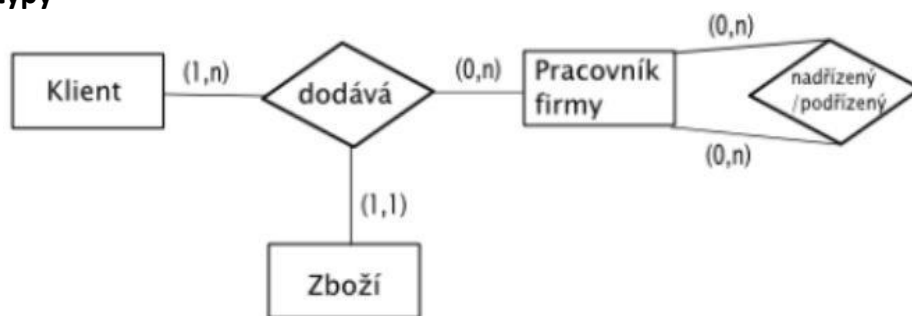
- každý ENTITNÍ TYP má nějaké atributy (např. Jméno), z nichž některé mohou být identifikátory (jednoznačně určují instanci entity)
- pokud ET nemá žádné identifikátory explicitně označené, jsou jimi všechny atributy dohromady (tzv. složený indikátor). Identifikátory mohou být více atributové



VZTAHOVÝ TYP

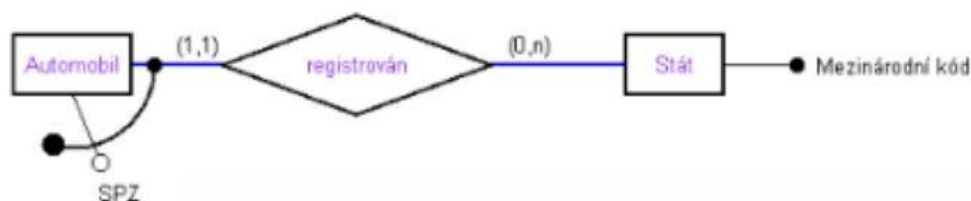
- VT (v diagramu ) popisuje vztahy mezi jednotlivými entitami – s těmi entitami, se kterými je v nějakém vztahu, je spojen čarou
- vztah může mít danou i KARDINALITU (kolik entit z každé strany do vztahu vstupuje), která může být typu 1:1, 1:n, n:n a je značená vedle čáry spojující vztahový typ s entitou
- entity ve vztahu mohou mít navíc povinné či nepovinné členství (vstupovat do něj vždy či někdy)
- vztahy mohou být buď binární nebo n-ární, ale více než ternární vztahy se většinou neobjevují. Vztahy mohou být i **REKURZIVNÍ**, tj. do vztahů vstupují entity stejného typu.
- instance vztahového typu je jednoznačně určena identifikátory instancí ve vztahu
- Některé entitní typy mohou být spolu identifikovány (nebo přímo identifikovány) vztahem – pak se nazývají slabé entitní typy.

Vztahové typy



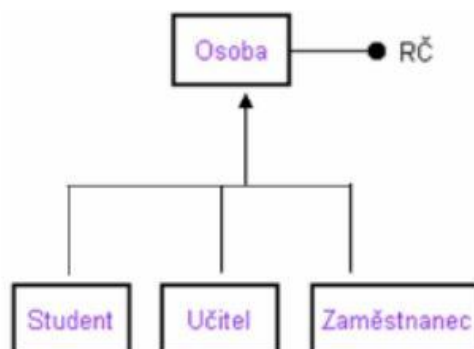
Obr. Ukazuje ternární vztah s různými kardinalitami – klientovi někdo dodává zboží jednou až n-krát, pracovník dodává nula až n-krát zboží (tj. jde o nepovinné členství ve vztahu, můžou existovat pracovníci, kteří nic nedodávají) a zboží je vždy dodáváno právě jednou. Na zaměstnancích je zároveň ukázán rekurzivní binární vztah.

Slabý entitní typ



Obr. Ukazuje slabý entitní typ – automobil je identifikován SPZ a zároveň státem, ve kterém je registrován.

ISA hierarchie



ISA hierarchie je rozšíření ER diagramů o dědičnost entit - tj. rozdělení entitních typů na subtypy (a přidání dalších vztahů nebo atributů pro subtypy). V ISA hierarchii se povoluje pouze jednonásobná dědičnost, navíc potomci nějakého entitního typu musí být jednoznačně identifikováni předkem (tj. všechny entity v hierarchii sdílí identifikátor).

Základní pojmy

Klíč (key) - jeden nebo několik atributů tabulky určený pro setřídění záznamů podle hodnot v tomto poli (numerický, textový)

Unikátní klíč - klíč tabulky, ve kterém se každá hodnota atributu vyskytuje nejvýše jedenkrát

Duplicitní klíč - klíč tabulky, ve kterém se každá hodnota atributu může vyskytovat vícekrát (u více různých záznamů)

Jednoduchý klíč

- klíč tabulky, který je tvořen pouze jediným atributem
- nejčastěji se vyskytující typ klíče
- i uměle vytvořený (id)

Složený klíč

- klíč tabulky, který je tvořen alespoň dvěma atributy
- často součást tzv. spojovací tabulky

Primární klíč (primary key)

- klíč tabulky, který slouží k jednoznačné identifikaci záznamu
- musí být unikátní
- označuje se PK
- v jedné tabulce může být nanejvýše jeden primární klíč

Cizí klíč (foreign key)

- klíč tabulky, který slouží k propojení (vytvoření relace) s primárním klíčem jiné tabulky
- často obsahuje duplicitní hodnoty
- označuje se FK
- v jedné tabulce může být i více cizích klíčů

ER model (postup vytváření)

• Určení typu entit

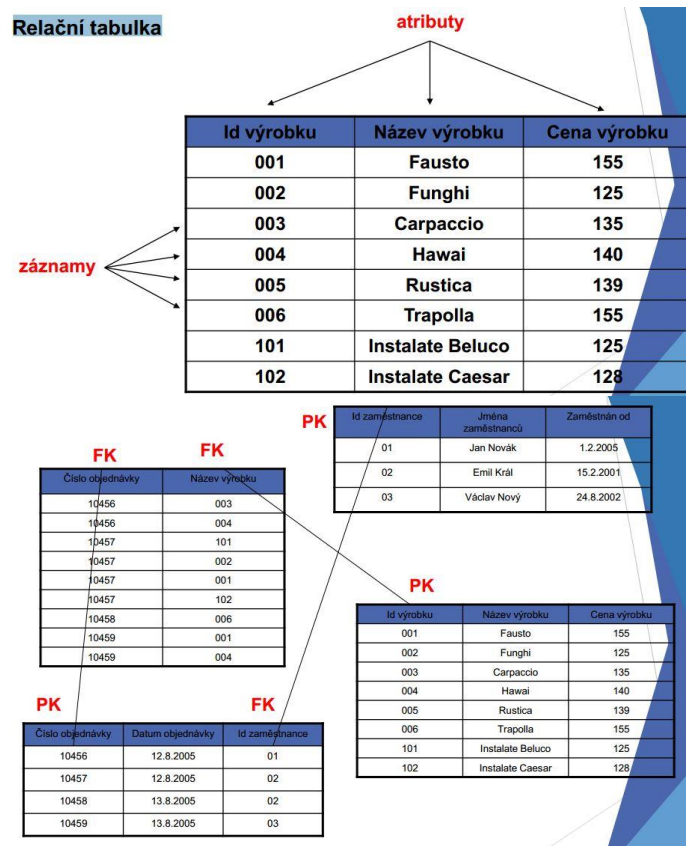
- zvolení množiny objektů stejného typu
- např. Objednávka, Zaměstnanec, Výrobek

• Určení typů relací

- vztahů, do kterých mohou příslušné entity vstupovat
- např. objednávka obsahuje výrobek

- **Určení atributů**
 - přiřazení jednotlivým entitám a vztahům
 - např. Objednávka (číslo, datum, ...)
- **Určení integritních omezení**
 - určení integritních omezení
 - - např. atribut datum je datového typu Datum a čas

ER model



QBE (Query By Example)

QBE = dotaz podle vzoru, je rozhraní pro zadávání dotazů grafickou formou. Znalost konkrétního dotazovacího jazyka není vyžadováno.

Lze provádět základní relační operace (selekce, projekce, spojení), ale také další operace jako třídění, součty, apod. (v závislosti na konkrétním SŘBD).

| Č. Studenta | Příjmení | Jméno | Adresa | Věk |
|-------------|----------|-------|--------|---------------|
| | √ No.. | √ | √ | √ >18, ≤25 |

Transakce

TRANSAKCE = posloupnost operací nad objekty BÁZE DAT, které realizují jednu nebo více ucelených operací z hlediska uživatele.

- transakce začíná vykonáním prvního příkazu nebo speciálním příkazem
- transakce může skončit úspěšně nebo neúspěšně
- neúspěch může vzniknout: poruchou hardware, chybou programového vybavení, chybnými daty, apod.
- bod, od kterého lze považovat transakci za úspěšnou se nazývá **bodem potvrzení**
- informace o změnách během transakce se ukládají do tzv. **žurnálového (transakčního) souboru**, teprve po dosažení bodu potvrzení se promítnou do báze dat
- při vlastní transakci se data v bázi dat nemění, tzn. při chybě nebo poruše nedojde k porušení konzistentního stavu báze dat

SQL (Structured Query Language)

SQL = strukturovaný dotazovací jazyk

- standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích
- vznik v 70. letech 20. století (IBM – výzkum relačních databází) jako sada příkazů pro ovládání RD – vznik jazyka SEQUEL (Structured English Query Language)
- cílem bylo vytvořit jazyk, ve kterém se příkazy tvoří syntakticky a v návaznosti na přirozený jazyk (angličtina)
- jazyk zahrnuje nástroje pro tvorbu databází (tabulek) a dále nástroje pro manipulaci s daty (vkládání dat, mazání, vyhledávání, atd.)
- SQL patří mezi tzv. DEKLARATIVNÍ PROGRAMOVACÍ JAZYKY, což znamená, že kód jazyka SQL nepíšeme v samostatném programu, ale vkládáme jej do jiného programovacího jazyka, který je již procedurální
- se samotným SQL můžeme pracovat pouze v případě terminálového přístupu na SQL server (skrže příkazový řádek)

Příkazy jazyka SQL

SQL příkazy se dělí do 4 zákl. skupin:

- **Příkazy pro MANIPULACI S DATY**
(např.: SELECT, INSERT, UPDATE,...)
- **Příkazy pro DEFINICI DAT**
(např.: CREATE, DROP,...)
- **Příkazy pro ŘÍZENÍ PŘÍSTUPOVÝCH PRÁV**
(např.: GRANT, REVOKE,...)
- **Příkazy pro ŘÍZENÍ TRANSAKČÍ**
(např.: START TRANSACTION, COMMIT,...)

SQL

SQL se skládá z několika částí:

- **DDL – Data Definition Language** – jazyk pro vytváření databázových schémat a katalogů
- **SDL – Storage Definition Language** – jazyk pro definici způsobu ukládání tabulek
- **VDL – View Definition Language** – jazyk pro návrháře a správce, určuje vytváření pohledů na tabulky
- **DML – Data Manipulation Language** – jazyk obsahující základní příkazy INSERT, UPDATE, DELETE, SELECT

S jazykem DML pracují nejvíce koncoví uživatelé a programátoři databázových aplikací.

SQL – uplatnění

SQL plní v současnosti celou řadu úloh, např.:

- **Interaktivní dotazovací jazyk** - uživatel zapíše příkazy SQL do interaktivního SQL programu, který najde potřebná data a zobrazí je na obrazovce
- **Databázový programovací jazyk** - k použití při tvorbě databázové aplikace
- **Jazyk pro správu databází** - správci databází využívají SQL pro definici datových struktur a kontrolu uložených dat
- **Klient/server jazyk** - programy pro počítače používají SQL pro komunikaci v lokální počítačové síti (LAN) s databázovými stanicemi (database servers), kde jsou uložena sdílená data
- **Jazyk distribuované báze dat** - systémy spravující distribuované databáze, používají SQL jako doplněk pro distribuci dat mezi několika spojenými počítačovými systémy
- **Komunikační jazyk SRBD v LAN** - v síti s několika různými SRBD je SQL jedinou cestou pro jejich vzájemnou komunikaci

4. METODIKA – VÝVOJOVÁ KLASIFIKACE (Metodiky, metody, techniky, nástroje)

Metodika

- je souhrn etap, přístupů zásad
- stanovuje – co, kdo, kdy a proč má dělat během procesu vývoje

Zahrnuje:

- organizace práce vývojového týmu
- metody práce s informacemi o vyvíjeném IS
- ekonomické otázky
- vedení projektové a provozní dokumentace
- způsob řízení v jednotlivých fázích vývoje IS
- SW a HW prvky doporučené pro vývoj IS

Metodika doporučuje použití určitých metod v průběhu vývoje IS, metody pak využívají určitých technik a nástrojů. Není však možné prohlásit, že daná metoda patří jednoznačně k určité metodice. Některé metody jsou specificky využívány konkrétní metodikou. Většina metod je univerzálních, využívají různé metodiky, v různých etapách vývoje IS.

Metodologie vývoje IS = zobecňující nauka o metodikách a metodách vývoje IS

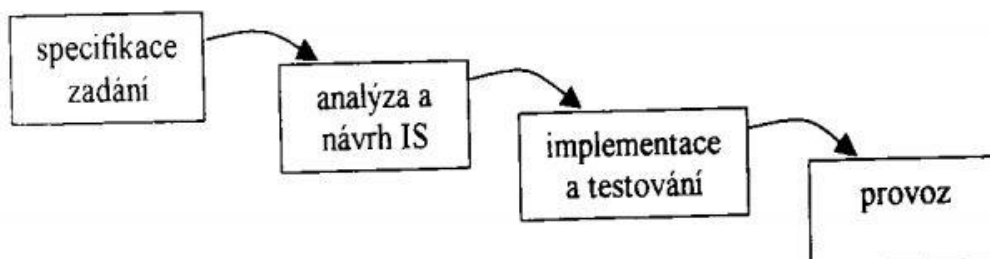
Metoda – určuje, co je třeba dělat v určité etapě vývoje IS. Bývá spojená s určitým přístupem (strukturovaný, objektový)

Technika – určuje, jak se dobrat požadovaného výsledku, tj. určuje přesný postup kroků, způsob použití nástrojů apod.

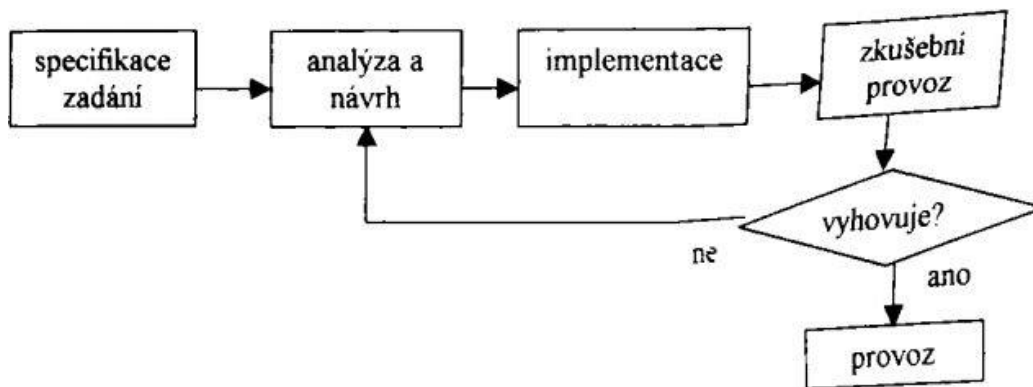
příklad technik: prototypování, normalizace datového modelu, transformační a transakční analýza při tvorbě struktury programového systému

Nástroj - prostředek k uskutečnění určité činnosti, resp. k vyjádření výsledku dané činnosti (formalizuje vyjádření výsledku). Může být svázan s konkrétní technikou, např. CASE nástroje, modely IS (datový, funkční, stavový diagram)

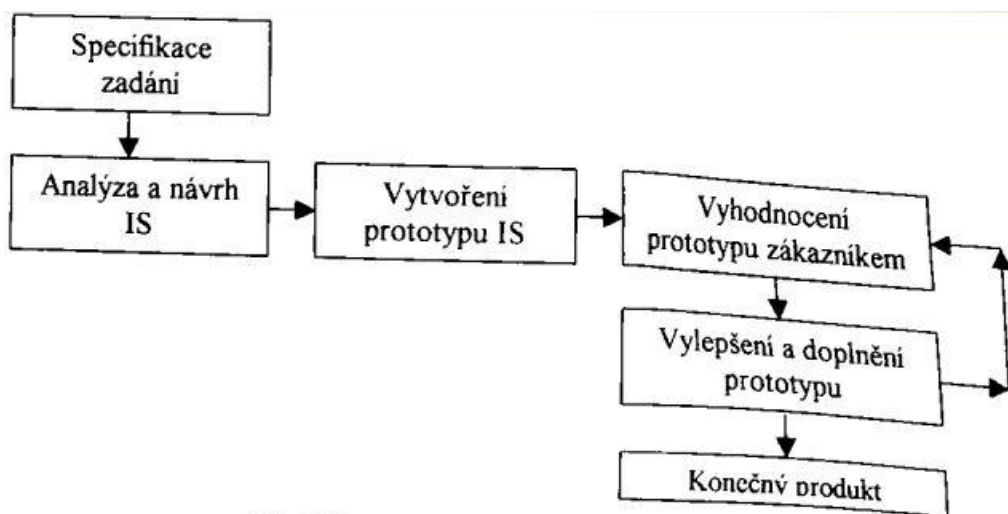
Životní cyklus IS



Obr 2.1 Model životního cyklu IS - "vodopád"



Obr 2.2 Model životního cyklu IS - "výzkumník"



Obr 2.3 Model životního cyklu IS - "prototyp"

Životní cyklus IS

Model „spirála“

- kombinuje prototypování s analýzou rizik. Jednotlivé etapy jsou cyklicky procházené vždy na vyšší úrovni podrobnosti analýzy, návrhu i implementace systému

Základní prostředky pro boj se složitostí vývoje IS

Hierarchický rozklad problematiky

- rozdělení složitého systému na subsystémy a to až do potřebné úrovně podrobnosti
- hierarchické rozdělení systému na subsystémy napomáhá plánovat, organizovat a kontrolovat práci vývojového týmu

Etapizace a iterace postupu řešení

- rozdělení složitého procesu vývoje IS na dílčí etapy. Každé etapě jsou přiřazeny cíle, úkoly, vstupy, výstupy, dokumentace, rizika, dílčí činnosti, odpovědné osoby, finanční náklady, apod.
- iterace znamená opakované provádění činností jednotlivých etap vždy na vyšším stupni porozumění problému. Účelem iterace je postupné zpracování problému na různých úrovních rozlišení – od hrubé představy o řešení až k podrobnému návrhu systému

Modelování a srovnávání modelů

- základní technika používaná během vývoje IS

Použití grafických vyjadřovacích prostředků

- umožňují vytvořit si názornou představu o vyvíjeném IS
- Grafické vyjadřovací prostředky jsou součástí CASE (Computer Aided System Engineering) tj. nástroje pro podporu vývoje IS – automatizují rutinní činnost

Analýza, návrh IS

a) Princip ABSTRAKCE

- myšlenkový proces, vylučuje odlišnosti a zvláštnosti jednotlivých objektů či jevů a zdůrazňuje společné, obecné, podstatné vlastnosti sledované množiny objektů či jevů

3 stupně ABSTRAKCE:

- kategorizace
- agregace
- generalizace

Kategorizace - nejnižší stupeň abstrakce, znamená seskupování prvků (jevů) do tříd (kategorií) podle kritérií, které si zvolíme k účelu sledování těchto prvků (jevů)

Agregace - je abstrakcí, při níž považujeme prvek za část většího celku. Jde o účelové sdružení prvků (tzv. abstrakce typu „část-celek“). Při agregaci nejde o zobecnění společných vlastností těchto prvků

Př. agregace prvků – komponenty pc: monitor, klávesnice, HDD – jde o prvky daného celku, tj. počítače.

Generalizace – abstrakce typu „specifický typ – obecný nadtyp“. Při generalizaci hledáme společné vlastnosti nadřazeného celku jakožto nositele specifikovaných společných vlastností (atributů)

Př. – společné vlastnosti prvků „správce počítače“, „operátor počítače“, „správce dtb. systémů“ - PRACOVNÍK SYSTÉMOVÉ PODPORY

Opakem GENERALIZACE je myšlenkový postup zvaný SPECIALIZACE.

b) KONKRETIZACE

- přístup, při němž postupně vyčleňujeme z obecného specifické vlastnosti sledovaných objektů či jevů
- opak abstrakce

Vývojová klasifikace IS

| Aspekt / typ IS | TPS | MIS | DSS | Expertné systémy | EIS |
|----------------------|---------------------------|-----------------------------|----------------------------|-------------------------------|--------------------------------|
| ÚČEL | len transakcie s dátami | najrôznejšie práce s dátami | pružné rozhodovanie | využitie znalostí experta | podpora vrcholového vedenia |
| VÝSTUPNÉ INFORMÁCIE | agregované správy | správy na vyžiadanie | informácie na rozhodovanie | hodnotenie, rady, vysvetlenia | kľúčové indikátory pre podniky |
| ORGANIZÁCIA DATABÁZY | jednotlivé súbory | súbory v interakcii | databázy a bázy modelov | databázy a bázy znalostí | napojenie: MIS a mimo podnik |
| PODPORA ROZHODOVANIA | slabá | pre rutinné úlohy | semištruktúrované problémy | aj neštruktúrované problémy | len orientačná |
| TYPICKÉ APLIKÁCIE | mzdy, evidencia materiálu | riadenie výroby, inventúry | strategické plánovanie | úvahy o investíciách | reakcia na okolie podniku |
| OBDOBIE VÝVOJA | 1955-1960 | 1962-1970 | pred r. 1980 | po r. 1980 | okolo r. 1990 |

Typologie IS

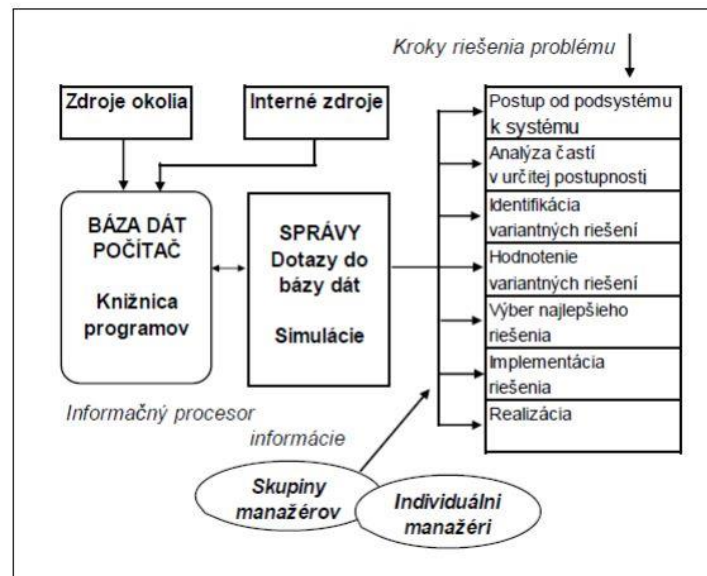
Průzkumové IS (Information Retrieval Systems) definované jako množinu lidí, technologií a procedur (software), které pomáhají vyhledávat údaje, informace a poznatkové zdroje lokalizované částečně v knihovnách nebo mimo ně. Informace o dostupných zdrojích jsou získávány, ukládány, vyhledávány a zpřístupňovány dle potřeb uživatelů.

Informační systémy pro podporu rozhodování (Decision Support Systems) jsou systémy se specifickými funkcemi orientovanými na pomoc manažerům při řešení problémů a v rozhodovacích procesech. Zahrnují lidi, procedury, software a účelové databáze.

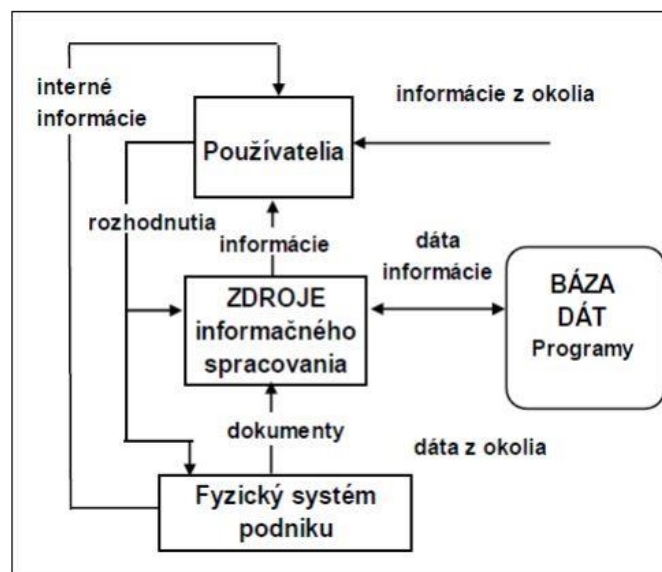
Pomáhají identifikovat faktory, které vytváří problémy; poskytují možné cesty řešení problémů; pomáhají vybírat možnosti, které jsou k dispozici k řešení problémů.

Expertní systémy (Expert Systems) jsou specifickým druhem informačních systémů, které pomocí software poskytují služby, které se očekávají od expertů. Jsou naprogramované imitovat myšlenkové postupy expertů a připravit návrhy rozhodnutí na výběr nejlepších partikulárních řešení problémových situací.

Manažerské informační systémy (Management Information Systems) zahrnují lidi, technologie a procedury, které slouží na organizační plánování, operační a řídicí přístup a využívání lidských a materiálních zdrojů.



Obr. DSS

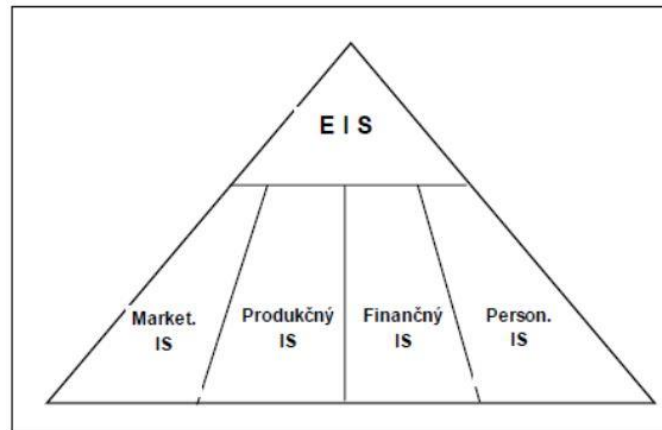


Obr. MIS

Systémy na přímé řízení technologických procesů. Jsou to systémy pracující v on-line-real-time (OLRT) režimu určené na přímé řízení technologických procesů, např. prostřednictvím NC strojů (numeric control) připojených na počítače.

Integrovaním přímého řízení procesů s organizací výroby, zásobování a expedice vznikají integrované výrobní informační systémy (Computer Integrated Manufacturing – CIM).

Informační systémy pro podporu vrcholového řízení (EIS – IS), které zabezpečují vrchol řídicí pyramidy, slouží především vrcholovému managementu podniku. Jsou to „osobní“ IS pro manažery na úrovni strategického plánování. Na rozdíl od MIS se EIS zajímá o informace z okolí podniku (technické inovace, trh, banka, konkurence apod.). EIS umožňují přístup k externím datům a sumarizují interní podnikové informace do nejvyšší úrovně agregace.

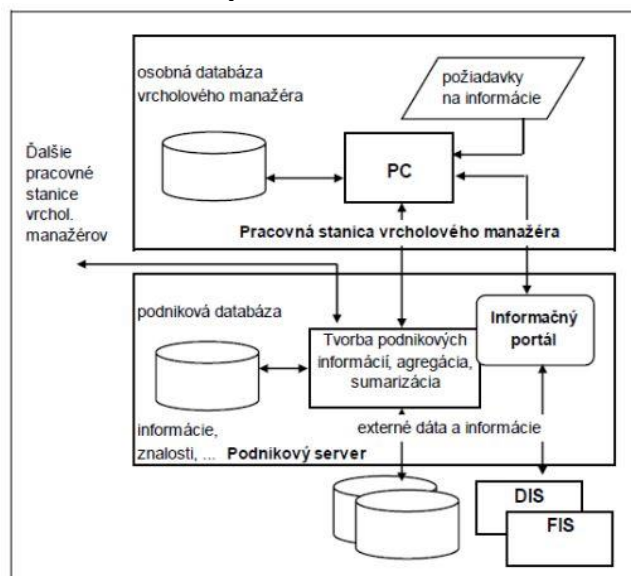


Obr. EIS

Srovnání MIS & DSS

| <i>Kritérium</i> | <i>MIS</i> | <i>DSS</i> |
|-------------------------------------|--|--------------------|
| <i>Hledisko podpory</i> | Organizace | Individuální |
| <i>Typ podpory</i> | Nepřímá | Přímá |
| <i>Podpora fází řešení problémů</i> | Identifikace, pochopení a realizace řešení | Všechny |
| <i>Typy podporovaných problémů</i> | Všechny | Samo strukturované |
| <i>Důraz na:</i> | Informace | Rozhodnutí |

EIS a jeho propojení na DIS a FIS přes IS



Podpůrné IS

1) Kancelářské IS (Office Automation – OA)

Obsahují:

- textové procesory
- faxy
- kopírovací přístroje
- zařízení na optické čtení dokumentů
- el. Poštu apod.

2) Útvarové systémy (Departmental Systems – DS)

- jsou často spojením TPS, DSS a OA, ale jejich rozsah je redukován na určitý útvar nebo místo

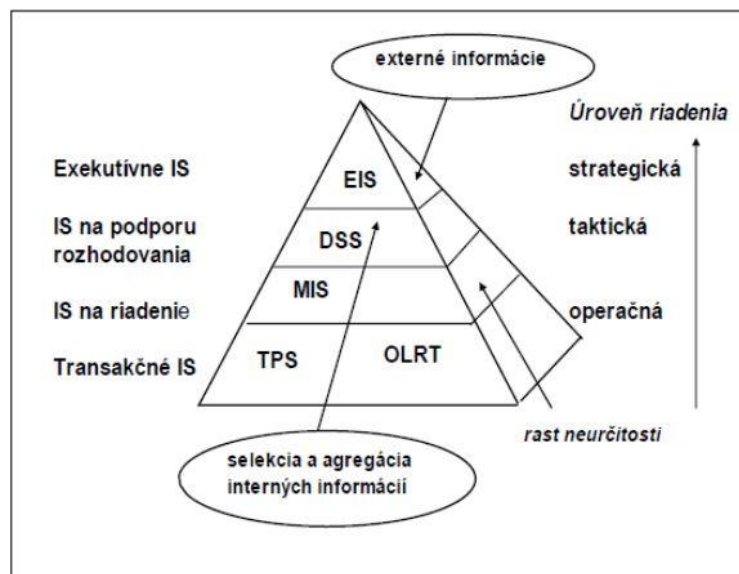
3) Dokumentografické (DIS) a faktografické (FIS) IS

- zpracovávají a poskytují odborné a vědecké informace sloužící k podpoře strategického rozhodování a plánování. Nejčastěji existují propojení z EIS na DIS nebo FIS přes informační portály.

Dělení IS dle obsahu výstupu

- agregované zprávy pro management (typické pro transakční IS)
- zprávy na vyžádání (Manažerské IS)
- informace pro rozhodování (IS na podporu rozhodování)
- hodnocení, rady, vysvětlení (expertní systémy)
- klíčové indikátory na řízení a strategické rozhodování v podnicích (exekutivní IS)
- adresy, příp. plné texty dokumentů (dokumentografické IS)
- fakta, souvislosti, sémantické mapy (znalostní a zpravodajské IS)

Dělení IS dle jejich vztahu k systému řízení



5. STRUKTUROVANÝ, OBJEKTIVĚ ORIENTOVANÝ PŘÍSTUP

Analýza a návrh IS

- Myšlenkové postupy ABSTRAKCE a KONKRETIZACE využíváme v průběhu celého procesu analýzy a návrhu IS.

Na myšlenkových postupech A a K jsou založeny principy:

- Princip rozlišovacích úrovní
- Princip tří architektur
- Princip modelování

Princip rozlišovacích úrovní

- Založen na zobrazení vyvíjeného systému na určité podrobnosti rozlišení detailů.
- V případě nejhrubší rozlišovací úrovně je IS znázorněn jako jeden prvek spolupracující s okolím – principem zobrazení je popis vazeb systému s okolím.
- Jde o víceúrovňové zobrazení systému (od rozlišovací úrovně „0“ až po úroveň „N“, na které vidíme detaily rozpracované do potřebné úrovně pro řešení problému.

Princip tří architektur

- Úzce souvisí s principem rozlišovacích úrovní.
- Veškerá zobrazení vyvíjeného IS jsou rozdělena do tří rozlišovacích úrovní – kategorií (vrstev podrobnosti) = tzv. VRSTVENÁ ABSTRAKCE SYSTÉMU.
 1. Vrstva – KONCEPTUÁLNÍ (esenciální)
 2. Vrstva – TECHNOLOGICKÁ (logická)
 3. Vrstva – IMPLEMENTAČNÍ (fyzická)

1. Vrstva zobrazení systému – KONCEPTUÁLNÍ

Zobrazení systému na hrubé rozlišovací úrovni – tj. zobrazení systému a jeho vazem na okolní systémy, dále zobrazení systému s viditelnými subsystémy a vazbami mezi nimi.

S takového zobrazení je patrné, které subsystémy fungují jako rozhraní mezi IS a okolím, resp. Které subsystémy zpracovávají VSTUPY a VÝSTUPY systému vzhledem k okolí.

ÚČELEM zobrazení 1. vrstvy je odpověď na otázku CO je obsahem systému?

2. Vrstva zobrazení systému – TECHNOLOGICKÁ

Zobrazení systému, která zviditelňují, **JAK je obsah systému realizován?**

Je zde zohledněna **organizace dat** (souborový systém, relační databázový systém, apod.) a **architektura systému** (např.: klient/server).

Zobrazení není zatíženo implementačními specifiky (tj. konkrétní realizace za pomoci prostředků ICT).

3. Vrstva zobrazení systému – IMPLEMENTAČNÍ

Detaily konkrétní implementace (např. vlastnosti plynoucí z konkrétního databázového systému).

Z této vrstvy je patrné ČÍM je systém realizován.

Princip tří architektur je aplikován v průběhu celého procesu analýzy a návrhu IS – vznikají MODEL Y vyvíjeného IS na jednotlivých úrovních ABSTRAKCE.

V jednotlivých etapách vývoje IS jde o snahu zpracovat model IS v požadované vrstvě.

Princip modelování

- V podstatě jde o TVORBU MODELU vyvíjeného IS.
- MODELOVÁNÍ = účelové zjednodušené zobrazení systému za pomoci vhodných (např. grafických) prostředků.
- MODELOVÁNÍ = ABSTRAKTNÍ obraz reality.
- MODEL = formalizovaný prostředek pro znázornění vyvíjeného IS, prostředek komunikace mezi odborníky, analytiky a uživateli IS. Znázorňuje strukturu systému (strukturu procesů, dat, atd.) na zvolené rozlišovací úrovni. Umožňuje optimalizaci struktury systému vzhledem ke zvoleným kritériím. Dále umožňuje simulaci s studium provedených změn systému a jejich vliv na subsystemy a okolí systému.

Charakteristika modelu

- Model formulován jako systém (tzn. znázorňuje prvky a jejich vzájemné vazby),
- Hraniční prvky realizují vazby s okolím systému (VSTUPY, VÝSTUPY),
- Obsah modelu je objektivní – každý prvek modelu odpovídá objektu reálného světa (tzv. pomocný prvek),
- uspořádání prvků modelu odpovídá uspořádání prvků části reálného světa, který model znázorňuje.

Analyzované dimenze IS

- **funkční** - popis procesů, datových toků a vazeb mezi subsystemy
- **datová** - popis druhů dat, se kterými bude IS pracovat
- **řídící** - popis časových souvislostí systémových akcí
- **organizačně-technologická** - popis a znázornění organizace práce s IS, popis zamýšlených provozních technologií, které bude IS realizovat
- **systémově-technologická** - popis realizace implementačně závislých systémových funkcí a jejich časové návaznosti

Druhy přístupu k analýze a návrhu IS

V průběhu historického vývoje se vyprofilovaly dva základní přístupy k analýze a návrhu IS:

- STRUKTUROVANÝ přístup (70. léta 20. stol.),
- OBJEKTIVĚ ORIENTOVANÝ přístup (90. léta 20. stol.).

Strukturovaný přístup

- Pojem „strukturovaný přístup“ odráží myšlenkový postup „strukturování“ (problematiky i předmětu zkoumání).
- V průběhu analýzy a návrhu IS potřebujeme zobrazit dva hlavní aspekty vyvíjeného IS:

- 1) PROCESY probíhající v systému
- 2) DATA, se kterými systém pracuje a která produkuje.

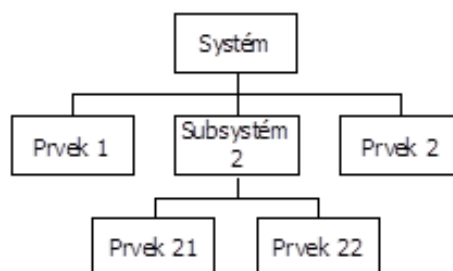
MODELY

STRUKTUROVANÝ PŘÍSTUP – charakteristické (na rozdíl od přístupu objektového) relativně samostatné zobrazení datových struktur systému v jednom modelu; datových toků a procesů zpracovávajících data v jiném modelu.

Metody strukturované systémové analýzy jsou plně v souladu s principy obecné teorie systémů

Funkční struktura

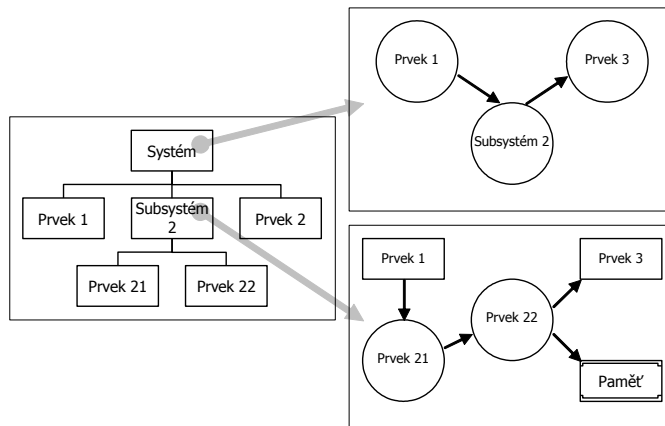
- Metoda analýzy funkční struktury je jednou ze základních metod strukturované analýzy používaná především k popisu struktury systému. Jedná se o metodu grafickou, která slouží k zachycení hierarchické dekompozice systému na subsystemy a prvky pomocí stromových diagramů.



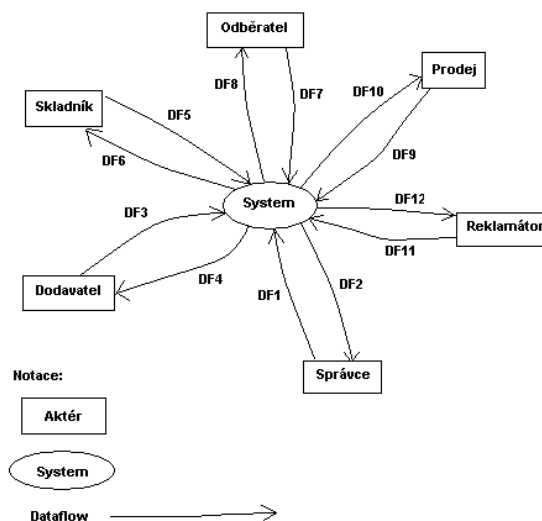
Informační toky

- Grafická metoda, která formou hierarchicky uspořádaných síťových diagramů vyjadřuje dekompozici systému na subsystemy a prvky a současně dovoluje zachytit informační vazby mezi těmito prvky.
- Vhodná metoda pro studium strukturálních vlastností systému.

- Základními aktivními prvky jsou funkční prvky neboli funkce, prvky zajišťující transformaci vstupní informace na výstupní.
- **Aktivní prvky** lze dále rozlišovat na prvky příslušné k popisovanému systému a prvky, které lze považovat vzhledem k popisovanému systému za vnější.
- **Pasivní prvky** představují paměti. Jedná se o prvky, které jsou schopny uchovat uloženou informaci. V případě softwarově orientovaných systémů mohou být realizovány například soubory nebo databázemi, v oblasti nesoftwarových systémů například protokoly, seznamy nebo záznamovými knihami.
- Významnou složkou diagramu informačních toků jsou **informační vazby mezi prvky systému** – informační toky. Obsah informačního toku nemůže být s ohledem na požadavek přehlednosti diagramu vyjádřen zcela detailně.
- **Metoda informačních toků** vyjadřuje formou hierarchicky uspořádaných síťových diagramů dekompozici systému na subsystémy a prvky a současně zachycuje informační vazby mezi těmito prvky.
- Na vrcholu této hierarchie stojí **tzv. kontextový diagram**, který vyjadřuje začlenění systému do souvislostí okolního světa.

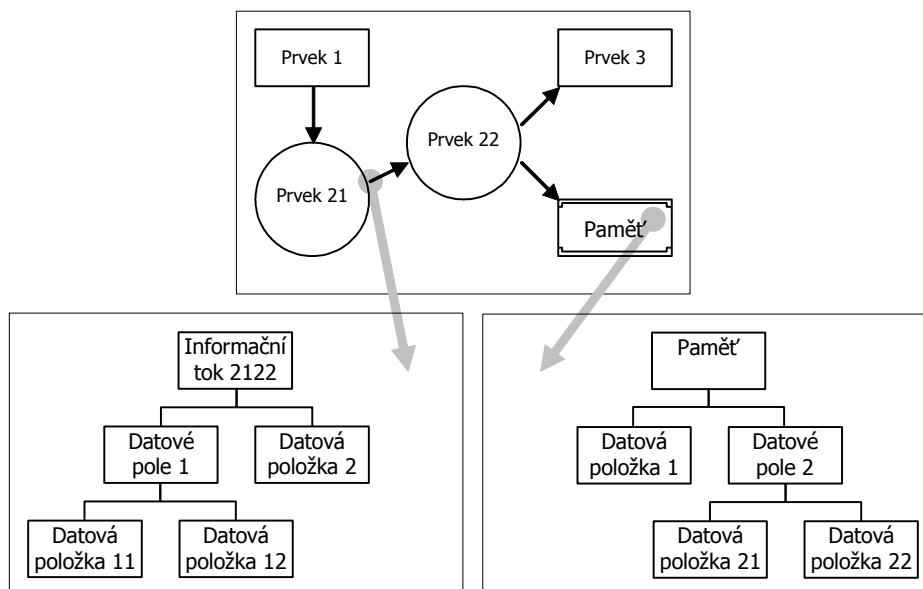


Kontextový diagram



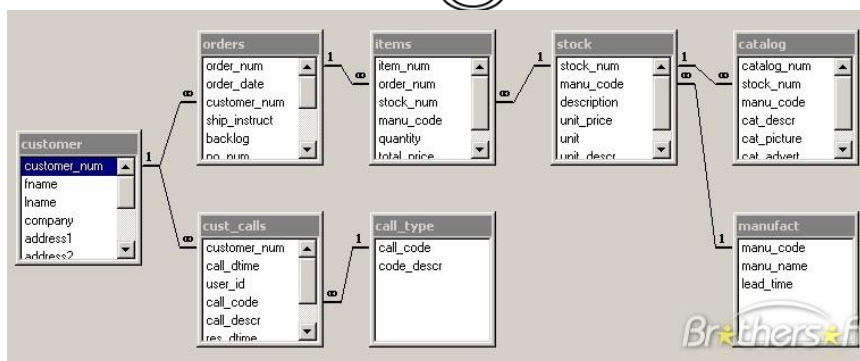
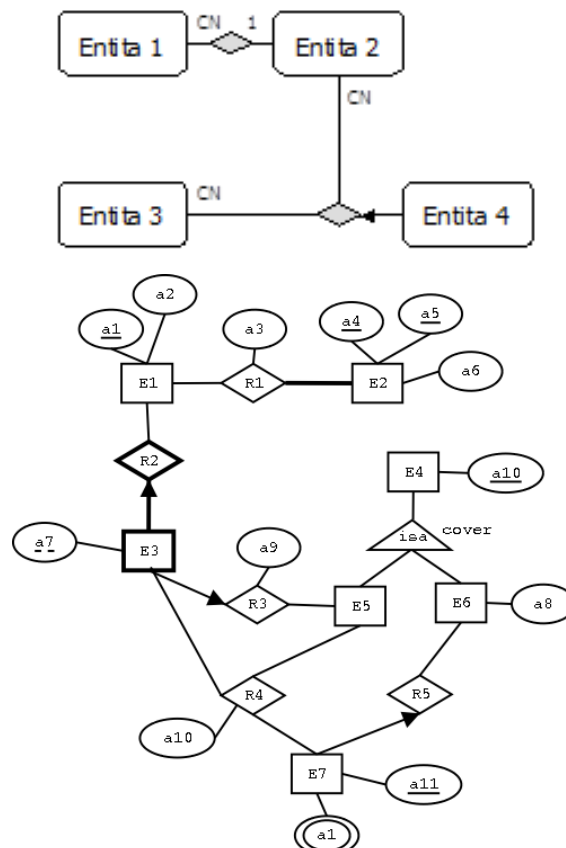
Datové struktury

- Metoda analýzy funkční struktury a metoda informačních toků neposkytují vhodné a dostatečné prostředky pro analýzu datové složky systému. Jak vazby mezi prvky systému popsané informačními toky, tak i pasivní prvky systému mohou představovat složité a rozsáhlé datové struktury.
- Jedním z prostředků datové analýzy je **popis datových struktur**. Analýza datových struktur umožňuje pomocí hierarchických stromových diagramů postupné rozčlenění informačních toků nebo paměťových prvků. **Všechny prvky tohoto diagramu představují obecně chápané bloky informace – datové položky.**
- Na analýzu struktury dat bezprostředně navazuje detailní datová analýza, která je prostředkem k podrobnému popisu elementárních datových položek – **listových prvků hierarchického stromového diagramu.**
- Detailní datová analýza umožňuje přiřadit každé elementární položce datové struktury datový element, který určuje formu uložení informace. Pro datové elementy je možno specifikovat například typ, rozsah nebo výčet možných hodnot.



ER model

- ER model – model entit a jejich vzájemných vztahů.
- Vhodný pro analýzu systému v případě, kdy složitost systému spočívá spíše ve složitosti struktury dat než ve složitosti jeho funkčních složek.
- ER model zachycuje formou síťového grafu objekty reálného světa a vztahy mezi nimi. Množiny objektů reálného světa mající shodné vlastnosti se nazývají entitami, vztahy mezi nimi pak relacemi. Entity mohou být blíže specifikovány množinou atributů, které mají shodný význam jako datové elementy užívané při detailní datové analýze.
- Relace mezi entitami jsou specifikovány kardinalitou a těsností vazby. ER modely se využívají pro tvorbu modelů dat na logické neboli konceptuální úrovni, tedy modelů dat nezávislých na jejich fyzické realizaci prostřednictvím specifického databázového systému.

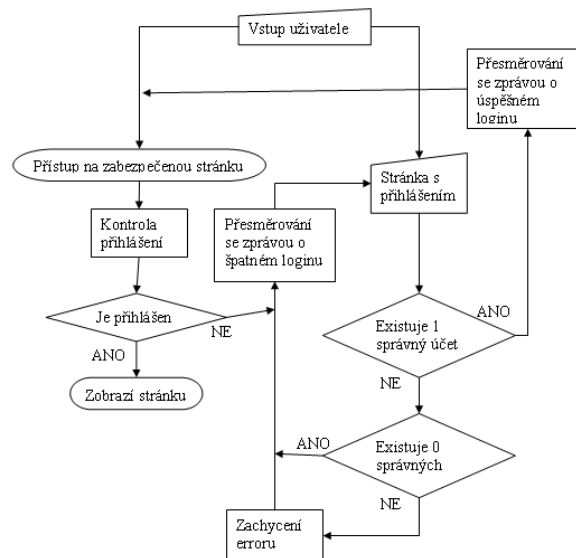


Metody popisu chování

- Popis chování systému je v obecném slova smyslu jeho algoritmizací.
- Algoritmus chování je však v určitých případech nutné nebo účelné podrobněji popsat již ve fázi analýzy. Metody popisu chování umožňují ve fázi analýzy zachytit algoritmickou složku systému, avšak abstrahují od konkrétního způsobu realizace, který je předmětem fáze návrhu.
- Metody jsou založeny na grafickém vyjádření nebo na vhodné kombinaci grafického vyjádření a formalizovaného textového popisu. Obvykle je pomocí grafických prostředků zachycena základní struktura algoritmu, která je pro specifikaci algoritmu na detailní úrovni doplněna relativně krátkými sekvencemi příkazů zapsanými formalizovaným jazykem.
- Klasickým grafickým prostředkem zápisu algoritmu je vývojový diagram. Díky své jednoduchosti získal oblibu v nejrůznějších oblastech, které daly vzniknout jeho různým modifikacím. Nejjednodušší varianta zachycuje základní strukturu algoritmu pomocí

vzájemně propojených elementů představovaných bloky a podmínkami. Detailní specifikace algoritmu je obsahem příslušných elementů a je vyjádřena formalizovaným jazykem.

Vývojový diagram



Metody popisu chování

- Metody popisu chování jsou obvykle velmi příbuzné metodám užívaným při tvorbě programů.
- Při nedodržení určitých pravidel mohou konstrukce zachycené pomocí vývojových diagramů odporovat zásadám strukturovaného programování, a tím mohou znesnadňovat případnou následnou programovou realizaci.
- Mezi metody, které podporují a dodržují zásady strukturovaného programování, patří metoda grafického zápisu algoritmů podle Jacksona, označovaná jako Jacksonovy diagramy. Základní struktura algoritmu je popsána hierarchickým stromovým diagramem, detailní specifikace je obsahem příslušných elementů a je vyjádřena formalizovaným jazykem.

6. PODNIKOVÝ INFORMAČNÍ SYSTÉM

Prvky IS

1 subsystém – lidé

- tvůrci (autoři) informací
- uživatelé informací (klienti)
- zpracovatelé, správci, zprostředkovatelé informací

2 subsystém – informace

1. Informace jako ekonomický zdroj

- IS jako jeden z pomocných subsystémů organizace (instituce, firmy), zaměřený na podporu její činnosti
- provozovatel: každá obchodní i neobchodní organizace

2. informace jako komodita (zboží)

- IS jako "produkční" systém organizace (instituce, firmy), jejímž základním produktem či službou jsou informace (v tom případě i tato organizace musí mít vlastní IS zaměřený na podporu vlastního řízení)
- provozovatel: sektor informačních služeb, informační průmysl

3. subsystém - prostředky umožňující práci s informacemi (informační infrastruktura)

- jazyky
- informační a komunikační technologie (hardware - počítače a periférie, síťové prvky, software)
- pracovní postupy, techniky a metody
- materiální zabezpečení (budovy...)

Typy IS

1. Informační systémy organizací (informace jako ekonomický zdroj)

- podnikové informační systémy (BIS - business information system, enterprise information system)

2. Veřejné informační systémy (informace jako ekonomická komodita)

- TV, rozhlas, tisk, zpravodajské agentury, knihovny, informační instituce

3. Státní informační systém

- informační systémy státní správy a samosprávy, informační systémy veřejné správy (GIS - government information system)

4. Osobní informační systém

- informační systém jednotlivce

IS organizací

Podnikový informační systém

- informační systém, provozovaný v kontextu konkrétní organizace
- **účel:** správa informací a znalostí a jejich integrace do podnikových procesů za podpory informačních a komunikačních technologií
- obsažené informace jsou chápány jako jeden z ekonomických zdrojů (aktiv) organizace

1. Podpora řídicích a administrativních funkcí (slouží vnitřním funkcím organizace)

řízení: definování strategických cílů, plánování, příprava rozpočtu

administrativa: správa a optimalizace firemních zdrojů - zaměstnanců a jejich činností, inventářů materiálu, přístrojů a vybavení, prostor, financí

- a) Systémy na podporu provozu (chodu) firmy** - provozní, transakční IS - ERP - enterprise resources planning
- b) Systémy na podporu rozhodování** - MIS - management IS, EIS - executive IS, BI - business intelligence
- c) Systémy na podporu plánování** - APS - advanced planning and scheduling, SCM - supply chain management, HR - human resources
- d) Systémy řízení vztahů se zákazníky** - CRM - customer relationship management

2. Podpora činností a služeb organizace (podporují účel, kvůli kterému organizace existuje)

- CA (computer aided) technologie (CAD, CAM, CIM, CASE...)
- e-biznys
- kancelářské systémy (office automation)
- systémy pro tvorbu a správu dokumentu (DTP - desktop publishing, DMS - document management system)
- workflow management
- automatizované knihovnické systémy, dokumentografické systémy
- expertní systémy
- GIS - geografické informační systémy

TPS a jeho složky

Transaction Processing System – Transakčně procesní systém

- Podpora hlavních činností na operativní úrovni řízení
- Odlišnosti dle zaměření organizace (bankovníctví, logistika, výroba, obchod, apod.)
- Patří zde i řízení zakázek, technické plánování výroby (tvorba projektové dokumentace), operativní řízení výroby, kontrola kvality produkce, atd.

| Blok | Název | Popis |
|------|---------------------------------|---|
| CIS | Customer IS | Bezprostřední styk se zákazníky (odečty spotřeby, fakturace, objednávky...) |
| RIS | Reservation IS | Rezervační systémy v dopravě, cestovním ruchu,... |
| GIS | Geographic IS | Kreslení, digitalizace map, vytváření územních celků, navigační systémy GPS,... |
| CIM | Computer Integrated Manufacture | Integrace výrobních procesů |
| CAD | Computer Aided Design | Konstrukční, návrhářské a projekční práce, technické výkresy, ... |
| CAM | Computer Aided Manufacture | Automatizovaná podpora řízení výrobních provozů |

ERP (Enterprise Resource Planning)

Historie ERP (Enterprise Resource Planning)

- 2. pol. 20. stol. – metoda **MRP** (Material Resource Planning) – upřesnění budoucí potřeby materiálu (kolik?, kdy?)
- snížení materiálových zásob (optimalizace, snížení pojistných zásob apod.)
- metoda **MRP** nebrala však v úvahu dostupnost kapacit ani žádné jiné vlivy ovlivňující výrobu = plánování materiálu nezajišťuje dostatečný pohled dopředu
- vylepšení metodologie prostřednictvím **MRPII** (Manufacturing Resource Planning)
- metoda **MRPII** nad rámec **MRP** (tj. potřeby materiálu) stanovit i předpokládanou potřebu kapacit (kdy?, kolik?)
- nebyla však zohledněna skutečnost, že kapacity jsou na rozdíl od materiálu výrazně limitovaným zdrojem (materiál mohu dle potřeby dokupovat, kapacity však nelze „nafukovat“)
- **MRPII** plánuje zdroje jako neomezené = neposkytuje efektivní nástroje pro dopracování plánu
- Zpracování dle konceptu **MRPII** je spojeno se sekvenčním postupem výpočtu (oddělená výpočtu materiálu od kapacit) tzn. nemohou být uplatněny optimalizační metody, taktéž časově náročná metoda
- úlohy **MRPII** byly implementovány do TPS či ERP systémů
- práce s daty probíhala prostřednictvím SQL (nevhodný jazyk pro takového úlohy)
- **MRPII** nesplnila očekávání – zjednodušuje podnikové zdroje a v nejlepších případech generuje „snad proveditelné plány“
- princip **MRP** a **MRPII** byl obvykle obsažen v základní funkcionalitě podnikových systémů typu ERP, které nastupují v 90. letech
- současný trend ve výrobě – APS (Advanced Planning System)
- APS pracují na základě kritériálních funkcí a jsou schopny na základě váhových koeficientů těchto kritérií optimalizovat výrobní tok
- APS disponují schopností okamžité reakce (nebo s velmi krátkým prodlením) resp. odpovědi na otázky typu „Co se stane, když...?“
- umožňuje návrh optimální varianty na základě změn váhových koeficientů u parametrů (časových, nákladových, kapacitních)

Charakteristika ERP

Nástroj pro plánování a řízení všech klíčových interních podnikových procesů na všech úrovních řízení (operativní, taktická, strategická) tzn. zpracování agend typu logistika, personalistika, výroba, ekonomika, atd.

Současná podoba tzv. **ERP II** neboli „**Extended**“ ERP = důsledek požadavků z podnikové praxe = nutnost těsnějšího propojení s:

- **Externími procesy** (bez definovaného vlastníka, řízení nemá management pod kontrolou (oblast CRM a SCM)
- **Procesy podporujícími vrcholové rozhodování** (EIS, OLAP, DW)

Základní požadavky na funkcionalitu ERP

- automatizace a integrace hlavních podnikových procesů
- sdílení dat, postupů (know-how) a jejich standardizace uvnitř podniku
- vytváření a zpřístupňování informací v reálném čase
- schopnost zpracování historických dat
- celostní přístup (holistický) k řešení ERP koncepce

Požadavky na přínos ERP

- realizace měřitelných přínosů v oblasti snižování nákladů v důsledku neefektivního řízení podniku
- realizace měřitelných přínosů v oblasti řízení podnikových procesů a dostupnosti v reálném čase

ERP je tedy **finančně orientovaný IS** pro určení a plánování podnikových zdrojů potřebných k přijetí, zhotovení, dodání a zaúčtování zákaznického obchodního případu = jádro celého podnikového IS.

- evidence kódu DPH zákazníka v rámci EU na výstupních dokumentech (VAT registration number)
- správné účtování na účty DPH při importu zboží a služeb (postup tzv. dvouřádkového záznamu o DPH – import DPH EU vstup, import DPH EU výstup)
- podklady pro výkaz INTRASTAT
- podklady pro výkaz udávající objem exportu v rámci EU za sledované období
- měnová tabulka, Euro
- přechod na euroměnu, obchodování v Euro
- zánik lokální měny, konverze na Euro

ERP v ČR

ERP lze rozdělit dle funkcionality na 2 základní typy:

- a) **All-in-One**
- b) **Best-of-Breed**

| ERP systém | Charakteristika | Výhody | Nevýhody |
|---------------|--|---|--|
| All-in-One | Schopnost pokrýt všechny klíčové procesy (výroba, distribuce, ekonomika, personalistika) | Vysoká úroveň integrace, dostačující pro většinu organizací | Nižší detailní funkcionalita, nákladná customizace |
| Best-of-Breed | Orientace na specifické procesy nebo obory nepokrývá všechny klíčové procesy | Špičková detailní funkcionalita, nebo specifická oborová řešení | Obtížnější koordinace procesů, nekonzistentnost v získávání informací, nutnost řešení více |

a) All-in-One ERP

Do kategorie All-in-One lze zařadit takové systémy, které může zákazník nasadit prostřednictvím jediného ERP projektu a pokrýt přitom všechny hlavní procesy.

Nabízí širokou škálu oborových řešení ověřených u zákazníků na celém světě. Vysoká funkcionality, vysoké pořizovací náklady.

Příklady: SAP, Peoplesoft, SSA Global, MS Navision, SSA MAX+, LCS Helios IQ, K2, KARAT, atd.

b) Best-of-Breed ERP

IS VEMA – zaměření na ekonomiku a personalistiku

IS FEIS – oblast ekonomiky, logistiky a obchodu (středně velké firmy)

7. TRENDY VE VÝVOJI ERP

Trendy ve vývoji ERP systémů

- Dřívější trend v oblasti ERP – dosažení maximální funkcionality
- S rozšiřováním funkcionality ale rostla složitost systémů

Integrace

- nynější trend označovaný jako ERP druhé generace
- ERP si ponechává pouze funkce, pro které byl primárně určen (tj. podpora podnikových procesů)
- Další funkce se řeší integrací specializovanými produkty (reportovací nástroje, oblast workflow, atd.)

Podpora uživatelů

- Do této oblasti patří např. nástroje Business Intelligence
- Základní úkol BI – monitorovat, analyzovat a plánovat podnikové procesy
- Analytické nástroje – na rozdíl od reportingu nezobrazují pouze stav hodnot
 - Snaží se odpovědět na otázky „proč se to tak stalo“ a „co se bude dít dále“

Monitorování stavu podniku

- Reporty
- Sledování klíčových ukazatelů výkonnosti (scorecarding) – okamžitý pohled na stav podniku v definovaných ukazatelích, založených většinou na porovnání plánu a skutečnosti

Plánovací procesy

- Tvorba finančních a obchodních plánů, rozpočty a plány investic (činnosti typické pro MIS systémy).
- Systémy mohou generovat plány na základě údajů z minulých období, provádět jejich extrapolaci podle zjištěných trendů, simulace variant rozpočtů.

Integrace se správou dokumentů (DMS systémy)

- Data jsou uložena strukturovaně v ERP systému, ale část se nachází i mimo -> integrace s nástroji pro správu dokumentů -> sledování oběhu dokumentů, archivace, verzování.
- Dle nedávných průzkumů jen 15 % zaměstnanců má přístup do ERP.

Podpora mobility zaměstnanců

- Pro zajištění větší mobility zaměstnanců roste potřeba **přístupu přes webové rozhraní, PDA, SmartPhone** apod.

Řešení CRM

- Vlastní modul v rámci ERP nebo integrace s CRM systémem.

- Problémem plnohodnotné náhrady CRM funkcemi ERP je princip ERP systémů a to požadavek na jednoznačnost popisu určité situace.
- Příliš komplexní řešení ERP má problém při nasazení v malých a středních podnicích, což začíná být zajímavý segment na trhu. Prostředí malých firem se dynamicky rozvíjí a je potřeba, aby se systém vyvíjel podle rozvoje podniku.

EAM

- Jednou z funkcí ERP systému je také komponenta **EAM (Enterprise Asset Management -správa podnikového majetku)**. S reportovacími a analytickými nástroji je EAM platforma pro optimalizaci výkonnosti podnikových aktiv. Používání informačního systému v oblasti údržby je v ČR výjimečné.
- Součástí ERP se tedy stávají funkce CRM, BI, e-business, webové portály nebo kolaborativní scénáře.

ERP – PŘÍNOSY

- Zefektivnění a zrychlení ekonomických procesů
- Centralizace dat – dostupnost přesných a konzistentních dat, sdílení dat
- Snížení chyb
- Úspory investic do IT (v dlouhodobém měřítku)
- Zvýšení bezpečnosti IS (bezpečnost dat)
- Rychlejší výstupy pro vedení firmy (nemusí se připravovat podklady)
- Podpora pro účetnictví (u nadnárodních – podle mezinárodních standardů)
- Zvýšení konkurenceschopnosti
- Zrychlení schvalování dat (např. plateb)
- Možnost propojení s dodavateli a odběrateli

NEDOSTATKY ERP

- Vysoká cena
- Další náklady – údržba, školení, rozšiřování
- Závislost na dodavateli

Důvody, kdy nechtějí uživatelé ERP používat:

- Aplikace se špatně ovládá
- Funkčnost neodpovídá potřebám

ERP – strukturovaná data

- **ERP jsou primárně systémy založené na databázi**, tj. předpokládají strukturovaná tabulkově orientovaná data.
- Pro nestrukturovaná data je lépe použít systémy pro správu a oběh dokumentů (DMS – Document Management System) a tyto integrovat s ERP.

MODELY DODÁNÍ ERP

- **On-premise model.** Aplikace je nainstalována na serverech organizace vlastníci ERP systém. Organizace musí mít vnitřní zdroje na provoz a údržbu ERP systému. Na upgradech, aktualizacích a úpravách systému se podílí sama organizace spolu s dodavatelskou firmou. Jedná se o nejběžnější model využívání ERP systémů.
- **On-appliance model** – forma SaaS, zákazník využívá jen některé moduly a platí jen za to, co využívá
- **On-demand model.** Tento model je znám také pod pojmy ASP (Application service provider) nebo SaaS (Software as a Service). Přestože mezi jednotlivými pojmy jsou rozdíly, tak hlavní společný rys je, že ERP systém je dodáván vzdáleně přes internet. O aktualizace a upgrady systému se stará dodavatel, který ERP provozuje na svých serverech. U tohoto modelu bývají větší obavy o bezpečnost a spolehlivost služby, protože organizace nemá přímou kontrolu nad správou ERP systému. Customizace systému se provádí pomocí tzv. mashupů.

MASHUP

- Mashup není produkt, služba nebo technologie, ale princip: **vytvářet nové služby integrací stávajících.**
- Liší se přidanou hodnotou, integruje se prostřednictvím API. Vytvoříme novou webovou službu nebo stránku s využitím webových služeb třetích stran. Příkladem je např. Použití GoogleMap v aplikaci na webu.

SaaS

- **SaaS (software jako služba)** vede k optimalizaci finančních toků (odpadá například nutnost velké počáteční investice). U správně implementovaného SaaS uživatel nepozná, že systém je hostovaný. V roce 2009 byl odhad trhu SaaS 10 mld USD.

Varianty SaaS

- Vlastní řešení pro každého zákazníka
- Konfigurovatelné řešení – separátní instalace se stejným aplikačním kódem
- Konfigurovatelné řešení pro více nájemců – v rámci jedné instance
- Konfigurovatelné rozšiřitelné řešení – víceúrovňová architektura, load balancing – proměnlivý počet serverů

VÝZNAMNÍ VÝROBCI ERP

- SAP
- Lawson
- Oracle Applications
- IFS
- Nexedi
- Infor
- ABAS AG
- Microsoft (Dynamics AX, Dynamics NAV)

VÝZNAMNÍ VÝROBCI ERP

Microsoft Dynamics NAV

- Produkt je součástí řady Microsoft Dynamics, která pomáhá firmám s účetnictvím a ekonomikou, řízením vztahů se zákazníky, dodavateli, provozní analytikou a e-komercí. Microsoft Dynamics NAV 2009 s novou třívrstvou architekturou klienta s novým uživatelským rozhraním zaměřeného na role (Role Tailored Client - RTC) byl uveden na trh v prosinci roku 2008.
- Do dalších verzí jsou plánovány nové funkcionality aplikace, klient pro SharePoint, implementace všech částí systému v .NET (a tedy podpora 64bit platformy a podpora Unicode) a další.
- Existuje dokument "Statement of Direction" Microsoft uveřejňuje pro své partnery a zákazníky, obsahuje směřování aplikace až do roku 2017.

Infor

- Získal formou akvizic celou řadu společností, čímž se stal třetím největším poskytovatelem podnikových aplikací (měřeno obratem) za SAP a Oracle Corporation. Dle různých firemních a mediálních zdrojů je Infor s obratem 2,2 miliardy dolarů dnes desátá největší softwarovou společností na světě.

SAP

„Systems - Applications - Products in data processing“ (Německo, Waldorf)

SAP R/3 se skládá z následujících modulů:

- FI (Financial Accounting) Finanční účetnictví
- CO (Controlling) Kontroling
- AM (Asset Management) Evidence majetku
- PS (Project systém) Plánování dlouhodobých projektů
- WF (Workflow) Řízení oběhu dokumentů

SAP R/3 se skládá z následujících modulů:

- IS (Industry Solutions) Specifická řešení různých odvětví
- HR (Human Resources) Řízení lidských zdrojů
- PM (Plant Maintenance) Údržba
- MM (Materials Management) Skladové hospodářství a logistika
- QM (Quality Management) Management kvality
- PP (Production Planning) Plánování výroby
- SD (Sales and Distribution) Podpora prodeje

SAP R/3 je client/server aplikace využívající třívrstvý model. Prezentační vrstva nebo klient komunikují s uživatelem. V aplikační vrstvě je uložena business logika a databázová vrstva zaznamenává a ukládá všechna data systému včetně transakčních a konfiguračních dat.

Funkčnost systému SAP R/3 je programována vlastním proprietárním jazykem ABAP (Advanced Business Application Programming, od 2003 je možné používat i Javu).

- ABAP (ABAP/4), je jazykem čtvrté generace (4GL) umožňujícím vytvářet jednoduché, ale výkonné programy. R/3 obsahuje také kompletní vývojové prostředí, které umožňuje vývojářům modifikovat existující programový kód SAPu nebo vytvářet vlastní funkčnost, od reportů až po transakční systémy, s využitím SAP frameworku. ABAP komunikuje s databází pomocí SQL dotazů, které umožňují vybírat, měnit a mazat data. Dále umožňuje vytvářet grafická uživatelská rozhraní a middleware pro integraci s jinými systémy.
- Informační systém pro **střední a malé firmy** (SMB - Small and Medium Business) jsou dodávány pod názvem **SAP Business One**.

OPEN SOURCE ERP

- Např. Compiere, JFire, OFBiz,...
- Volná dispozice zdrojovým kódem
- Možnost změnit užití SW dle budoucích potřeb
- Nevýhody Open Source řešení:
- Nestálost v čase
- Nejasná koncepce vývoje
- Může být nedotaženost projektu – lokalizace atd.
- Komerční řešení jsou více zaměřena na implementační fázi.

TRH S ERP SYSTÉMY

- Large Enterprises – nadnárodní společnosti – segment již obsazen.
- Medium Enterprises – pro dodavatele nejzajímavější segment
- Small Enterprises (do 50 zaměstnanců a 100 mil. Kč) – krabicové produkty tuzemských výrobců
- Menší firmy kladou důraz na rychlost a snadnost pořizování dat na úkor šíře, komplexnosti a kvality.
- Čím menší firma, tím nižší nároky na zpětné vyhodnocení dat, naproti tomu má vyšší nároky na rychlost a jednoduchost pořízení vstupních dat. To je v rozporu s původním požadavkem na kvalitu, šíři a komplexnost dat.

Proto menší firmy nedávají přednost komplexním velkým balíkům, kde je důraz na komplexnost, kvalitu a špičkové analytické informace; pro malou firmu se implementace takového balíku může stát komplikací.

8. TESTOVÉ OTÁZKY

První test

Co je to informační systém + nakresli konceptuální model?

Jak funguje centrální architektura systému?

Co je to systém řízení báze dat?

Jaká data zpracovává IS?

Jaké jsou relace mezi entitami?

Co je to slabý entitní typ?

Jaké jsou typy systémů?

Co je to redundance?

Jaké jsou datové typy + uveď příklady?

Kdo je autorem teorie disipativní struktury?

Druhý test

Vyjmenuj životní cykly související s návrhem a vývojem IS.

K čemu slouží expertní IS?

Popiš 3 stupně abstrakce.

Vysvětli princip 3 architektur.

Uveď základní charakteristiky modelu.

Vyjmenuj základní prostředky pro boj se složitostí systému.

K čemu slouží Diagram nasazení?

Charakterizuj strukturovaný přístup.

Vysvětli vztah entita-atribut-instance

K čemu slouží CASE nástroje?

Třetí test

K čemu slouží Transaction Processing Systém (TPS), uveď příklady užití v organizaci

Charakterizuj současnou podobu ERP (Extended ERP)

Uveď rozdělení ERP dle funkcionality včetně základní charakteristiky.

Jaké jsou současné trendy ve vývoji ERP?

Charakterizuj službu SaaS

Uveď příklady veřejných IS

Vysvětli termín „implementace“

K čemu slouží CRM systémy (customer relationship management)?

Jaké jsou hlavní nedostatky ERP systémů?

Co je to SAP? Stručně charakterizuj.