

Teoretické základy projektování IS

KISK FF MU

VIKMB46

Jarní semestr 2014/15

Mgr. Jan Matula, PhD.

jan.matula@fpf.slu.cz

Cíle předmětu

- Cílem předmětu je seznámit posluchače s teoretickými základy projektování informačních systémů, se základními principy fungování IS. Pozornost je také věnována analýze, projektování a zabezpečení IS. Předmět rozšiřuje znalosti zejména v oblasti modelování a administrace projektů IS. Na konci tohoto kurzu bude student schopen: porozumět a vysvětlit základní principy teorie systému; vytvořit projektový úkol a navrhnout IS.

Ukončení předmětu

- Odevzdání dílčích úkolů
- 50 % účast na přednáškách
- Účast na praktickém workshopu (termín bude upřesněn)

Realizace výuky

21. 2. 2015

14. 3. 2015

4. 4. 2015

16. 5. 2015

Doporučená literatura (Bc. studium)

- SODOMKA, P. Informační systémy v podnikové praxi. 1. vyd. Praha: Computer Press, 2006. 351 s. ISBN 80-251-1200-4.
- DOUCEK, P. Řízení projektů informačních systémů. 2. vyd. Praha: Professional Publishing, 2006. 180 s. ISBN 80-86946-17-7.
- KAJZAR, D. – POLÁŠEK, I. Projektování informačních systémů. 1. vyd. Opava: Ediční středisko FPF SU v Opavě, 2003. 219 s. ISBN 80-7248-214-9.
- VLASÁK, R. – BULÍČKOVÁ, S. Základy projektování informačních systémů. 1. vyd. Praha: Karolinum, 2003. 144 s. ISBN 80-246-0727-1.
- KIMLIČKA, Š. Princípy informačných systémov. Bratislava: STU, 2006. 250 s.

Doporučená literatura (specializace Nav. Mgr.)

- ŠKUTOVÁ, Jolana. Projektování informačních systémů: učební text předmětu Informační systémy [DVD-ROM]. Vyd. 1. Ostrava: Vysoká škola báňská - Technická univerzita Ostrava, 2011. Požadavky na systém: Adobe Acrobat Reader, Flash Player. ISBN 978-80-248-2766-7. Dostupné také z: <http://projekty.fs.vsb.cz/147/ucebniopory/978-80-248-2766-7.pdf>.
- ŠIMONOVÁ, Stanislava, MYŠKOVÁ, Renáta a JIRAVA, Pavel. Projektování informačních systémů - UML, procesní řízení: pro kombinovanou formu studia. Vyd. 1. Pardubice: Univerzita Pardubice, 2006. 114 s. ISBN 80-7194-895-0.

Doporučená literatura (specializace Nav. Mgr.)

- KALUŽA, Jindřich. Informační systémy pro strategické řízení. 1. vyd. Ostrava: Vysoká škola báňská - Technická univerzita, Fakulta ekonomická, ©2010. 145 s. ISBN 978-80-248-2280-8.

Sylabus předmětu

1) Rekapitulace základní terminologie IS a databází

2) Analýza a návrh IS

- Metodiky, metody, techniky, prostředky pro boj se složitostí systému, analýza systému, životní cykly, myšlenkové procesy princip abstrakce a konkretizace, 3 stupně abstrakce, princip rozlišovacích úrovní, princip tří architektur, princip modelování.

3) Strukturovaný a objektově orientovaný přístup (OOP)

- Funkční struktura, Informační toky, Kontextový diagram, Datové struktury, ER-modely, Metody popisu chování, Vývojové diagramy, Stavové diagramy, Metoda sekvenčních funkčních grafů, Objektově orientované modelování, výhody a nevýhody OOP, Historie OOP – vznik UML, UML – skupiny artefaktů, vztahy, diagramy (Use case, sekvenční diagram, Diagram spolupráce, Diagram tříd, Stavový diagram, Diagram aktivit, Diagram nasazení).

Sylabus předmětu

4) Přehled metod vývoje IS

- Přehled strukturovaných metod, historický úvod ke strukturovanému přístupu, Přehled objektově orientovaných metod.

5) CASE nástroje

- Historie a vývoj CASE nástrojů, obecné vlastnosti CASE nástrojů, možnosti využití nástroje MS Visio 2013 a dalších nástrojů (včetně volně šiřitelných)

6) Provádění analýzy a návrhy IS v podniku

- Metody a přístupy k analýze systému, Hrubá analýza a návrh systému, Specifikace požadavků na systém, Katalog požadavků, detailní analýza a návrh systému, Prototypování, Prezentace analýzy a návrhu IS, Vyjednávání během vývoje IS, zákaznická customizace.

Sylabus předmětu

7) Testování IS

- Základní pojmy procesu testování, fáze procesu testování, Plán a návrh testů, Příprava, provádění a hodnocení testů, Zjištění vad a neshod v systému, Zátěžové testování systému, Akceptační testy, testovací CASE nástroje.

8) Implementace IS do provozu

- úvod a základní terminologie, etapizace procesu zavádění IS, Pracovní role v procesu zavádění IS do provozu

9) Dokumentace k IS

- Standardizace dokumentace k podnikovým IS, Požadavky na administrátorskou dokumentaci k podnikovému IS.

Sylabus předmětu

10) Řízení projektů vývoje IS

- Základní pojmy, Zdroje pro zpracování projektu, Životní cyklus projektu, Organizace řízení projektu, Plánování projektu, Dekompozice projektových činností, Rozhodovací analýza, Síťová analýza, Řízení rozsahu projektu, Plánování času a využití časových rezerv, Plánování finančních nákladů, Řízení rizik, Dokumentace řízení projektu, Závěr projektu.

11) Systémová integrace

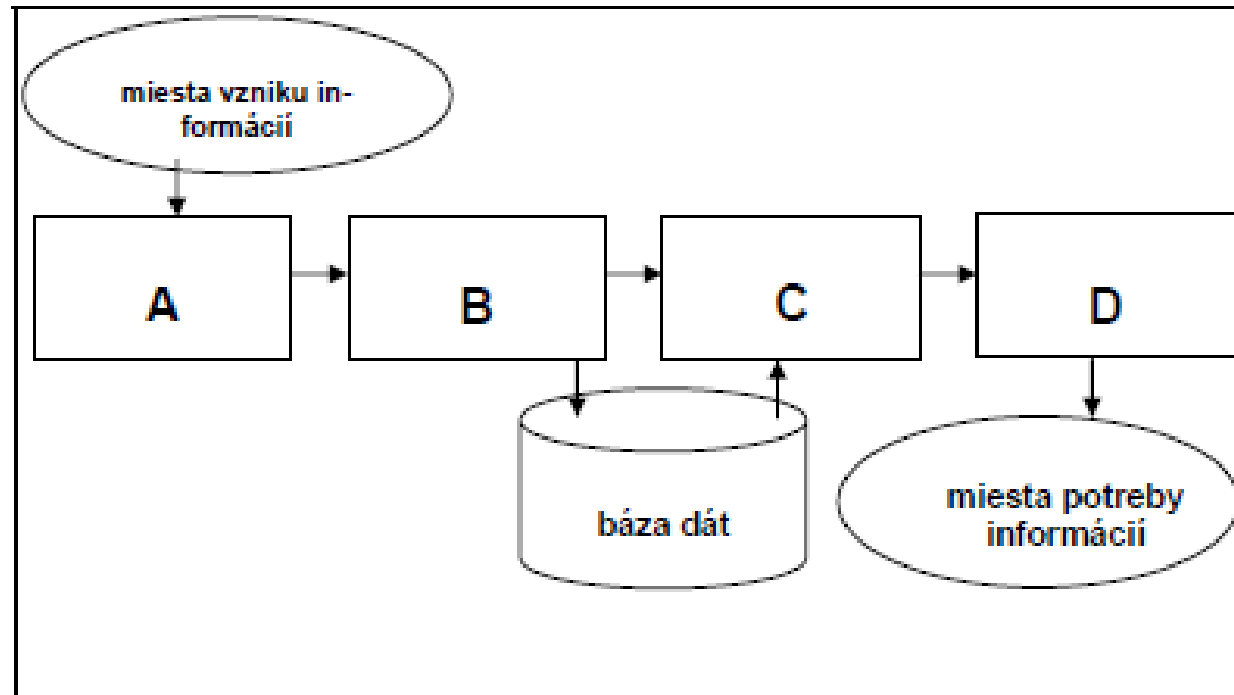
- Podstata systémové integrace, Vývoj a úrovně systémové integrace, Systémový integrátor, Outsourcing provozu IS, Poskytování aplikačních služeb IS.

Úvod do IS

Obecná charakteristika IS

- systém umožňující komunikaci a transformaci informací - časově, 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



Obr. 1.16 – Konceptuálny model informačného systému

Jaké informace zpracovává IS?

- **strukturované data popisující neprostorové objekty** (záznamy v databázích, souborech a pod.) – 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 a pod.),
- **metadata** (popis dat pomocí SGML jazyků – HTML, XML, struktury typu MARC, Dublin Core a pod.), 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

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.

Agendové zpracování dat

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.

Agendové zpracování dat

PROBLÉMY AGENDOVÉHO ZPRACOVÁNÍ

- **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.

Agendové zpracování dat

POUČENÍ Z AGENDOVÉHO ZPRACOVÁNÍ

- Vedlo k návrhu a vytvoření programových systému (systému řízení báze dat – SŘBD) následujících vlastností:
- 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
- existují prostředky pro definování všech **sledovaných vlastností** popisovaných objektu;

Agendové zpracování dat

POUČENÍ Z AGENDOVÉHO ZPRACOVÁNÍ

- existuje **soubor instrukcí**, které nad definovanými daty provádějí jednotlivé operace; každá instrukce je vlastně mohutnou procedurou, v níž je řešen fyzický přístup k datum i realizace vlastní operace; jinak než prostřednictvím systému není možno s daty pracovat;
- programový systém řeší způsob, jak zaznamenat **vztahy mezi objekty**.

Úvod do terminologie - pokračování

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.

Úvod do terminologie

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.

Úvod do terminologie

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.

Úvod do terminologie

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.

Úvod do terminologie

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.

Relační model - rozšíření

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.

12 pravidel pro relační SŘBD

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.

12 pravidel pro relační SŘBD

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.

12 pravidel pro relační SŘBD

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:

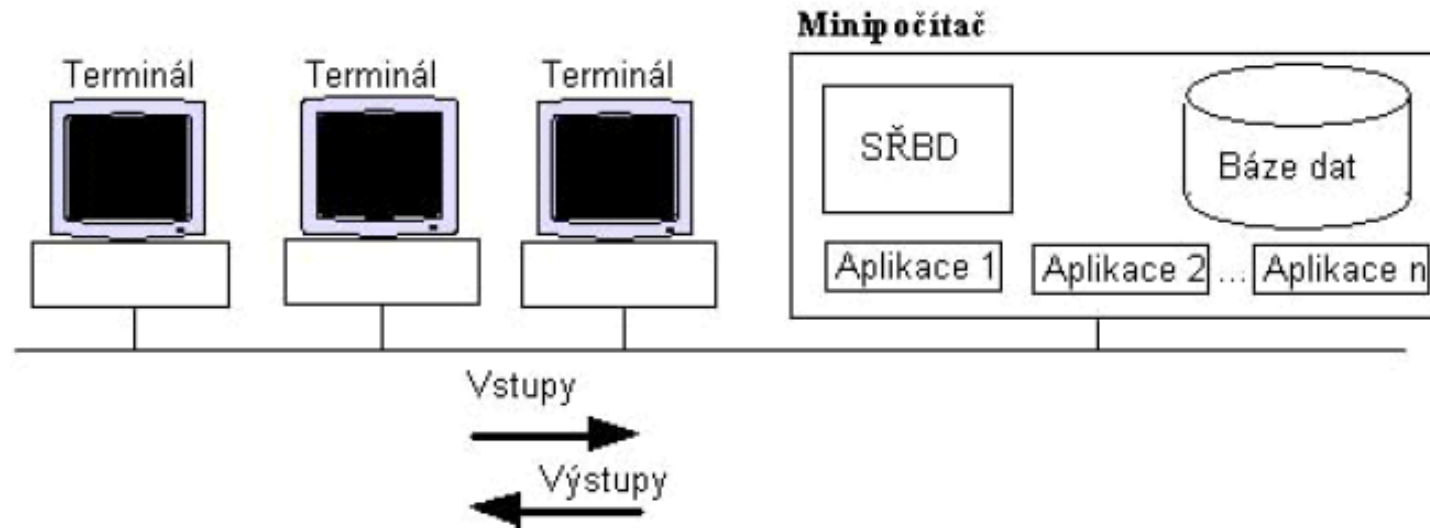
- **Centrální architektura**
- **Architektura file – server**
- **Architektura klient - server**

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.

Obr. Centrální architektury



Architektura file - server

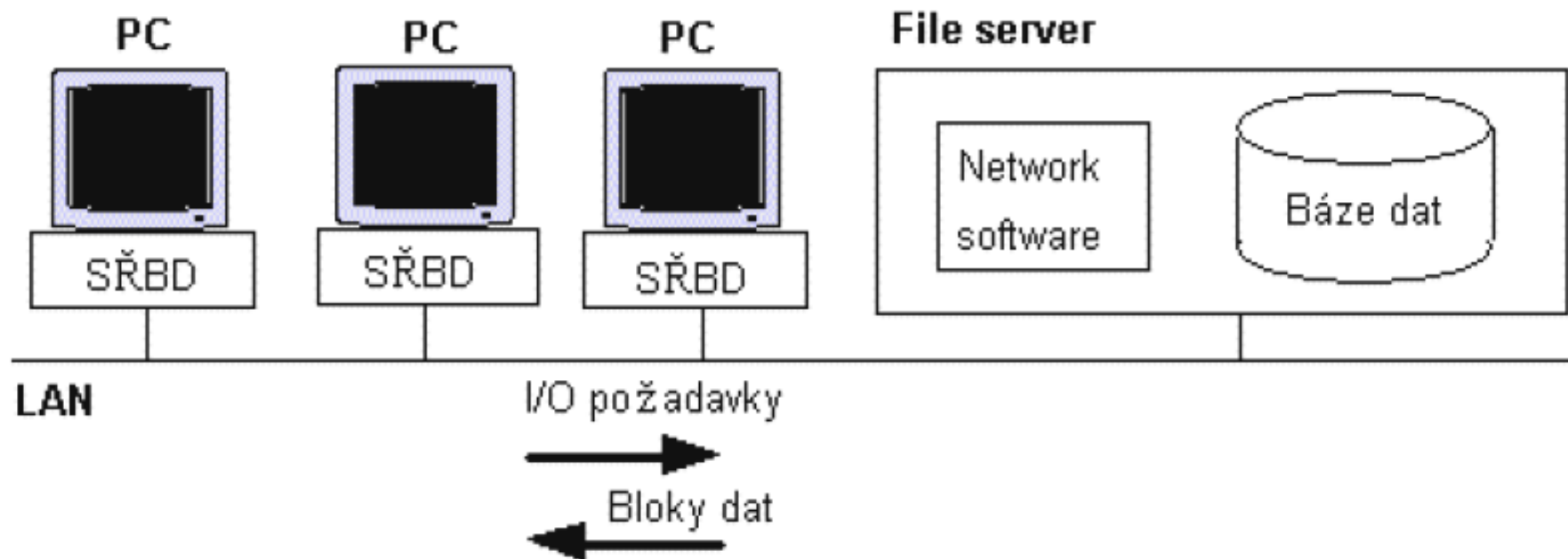
- Tato metoda souvisí zejména s rozšířením osobních počítačů a sítí LAN. SŘBD a příslušné databázové aplikace jsou provozovány na jednotlivých počítačích, data jsou umístěna na file-serveru a mohou být sdílena. Aby nedocházelo ke kolizím při přístupu více uživatelů k jednomu datům, musí SŘBD používat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

Architektura file - server

Komunikace uživatele se systémem probíhá následujícím způsobem:

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

Obrázek architektury file - server



Architektura klient - server

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.

Architektura klient - server

Komunikace probíhá tímto způsobem:

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

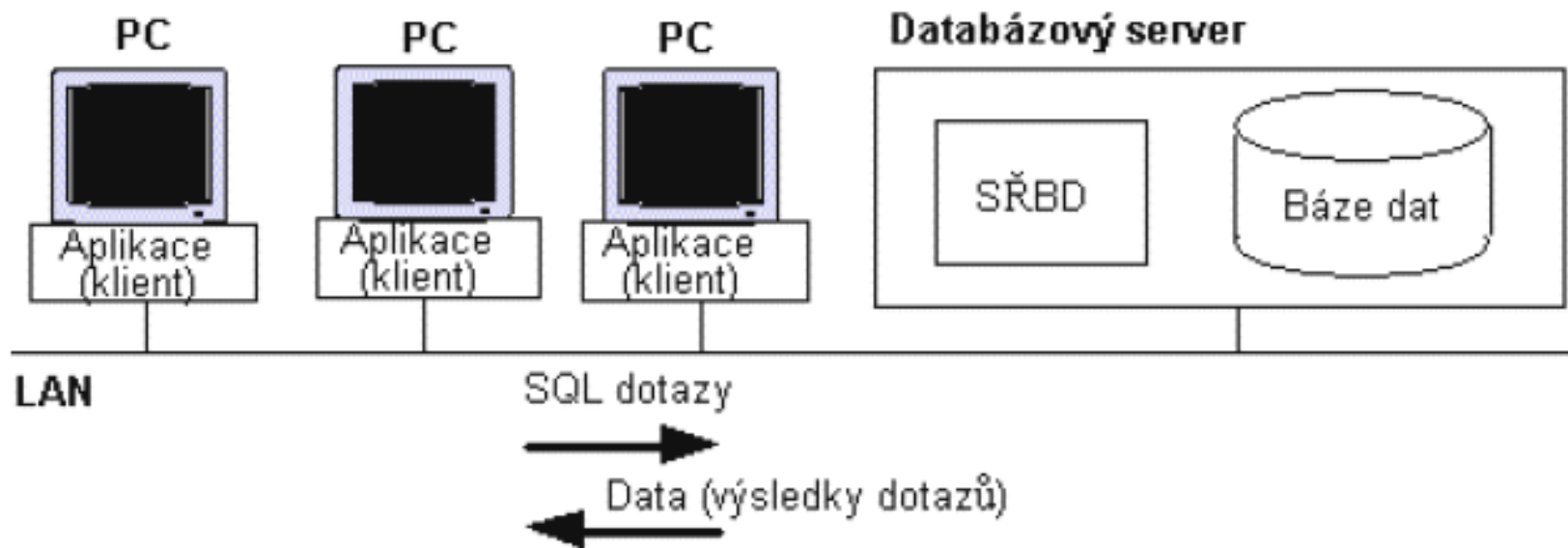
Architektura klient - server

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.

Obrázek architektury klient - server



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, nespecifikuje 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.

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**.

Základní pojmy DTB zpracování

SYSTEM ŘÍZENÍ BÁZE DAT – SŘBD

programový systém (prázdný, bez datových souboru a bez aplikačních 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.

Základní pojmy DTB zpracování

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

Vztahy mezi entitami (relace)

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.

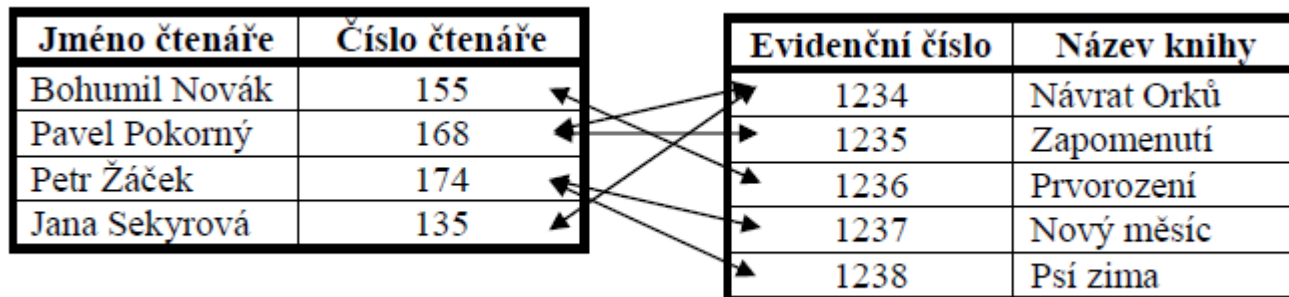
Vztahy mezi entitami (relace)

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

Vztahy mezi entitami (relace)

N:N



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

Hierarchický DM

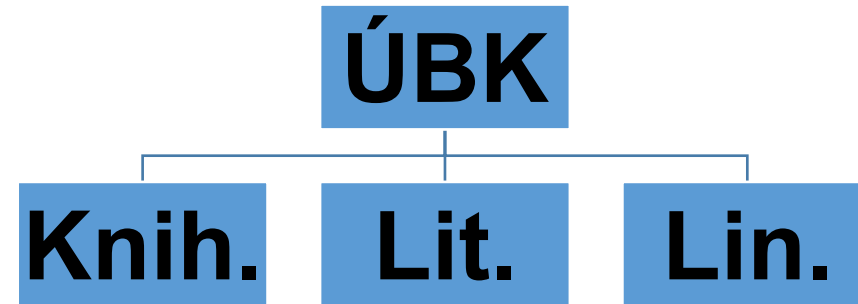
Záznamy jsou organizovány ve stromové struktuře

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



Síťový DM

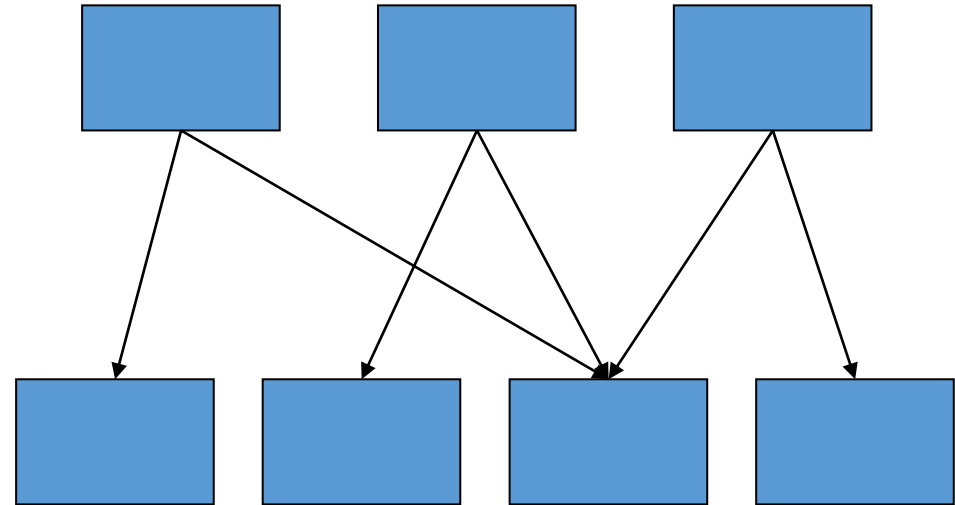
Data jsou reprezentována kolekcemi záznamů a vztahů mezi nimi.

VÝHODY:

- Řeší snadno a rychle vztahy 1:N i M:N
- Nezáleží na fyzické struktuře dat
- Rychlé vyhledávání

NEVÝHODY:

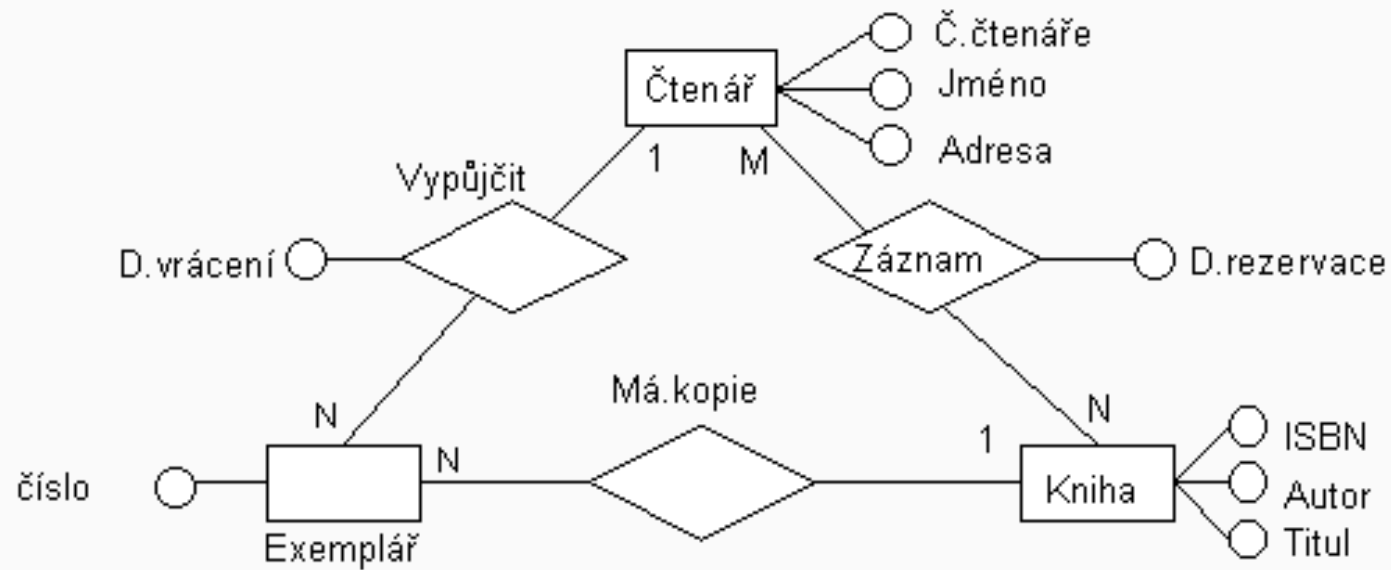
- Problémy při změně struktury dat



Relační DM

- **RDBS** – relační databázový systém
 - Informace uchovávány v jednom typu objektu jsou uchovávány 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


Objektový DM

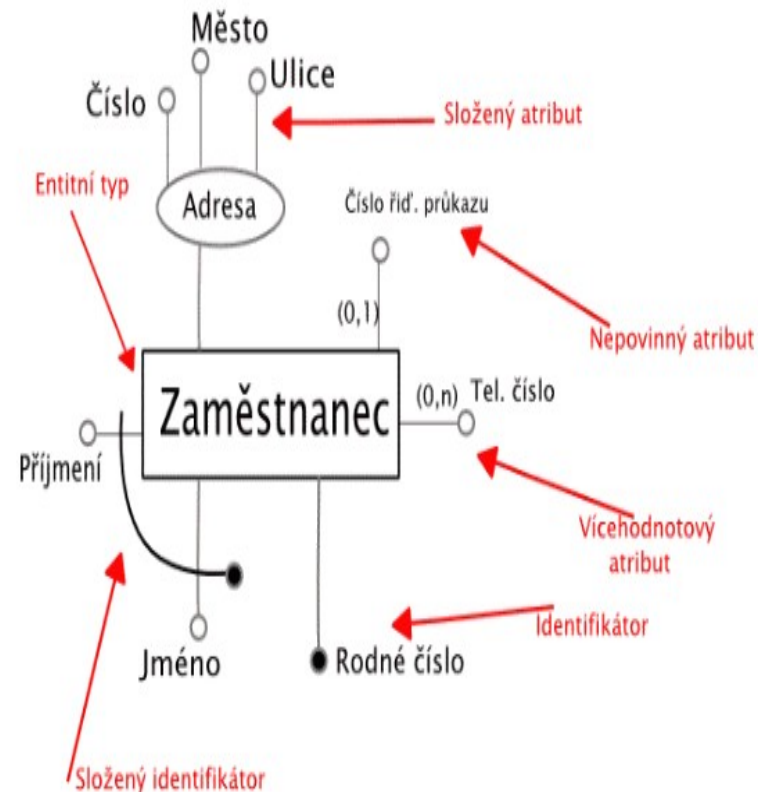
- Vychází z principů objektově orientovaného přístupu
- Objekt – datová struktura definovaná jako třída s určitými vlastnostmi a metodami
- Komunikace mezi objekty probíhá pomocí zpráv
- Výhody
 - Nejen statické, ale i dynamické chování objektů
 - Možné vytváření složitějších objektů
 - Snadnější zadávání 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.



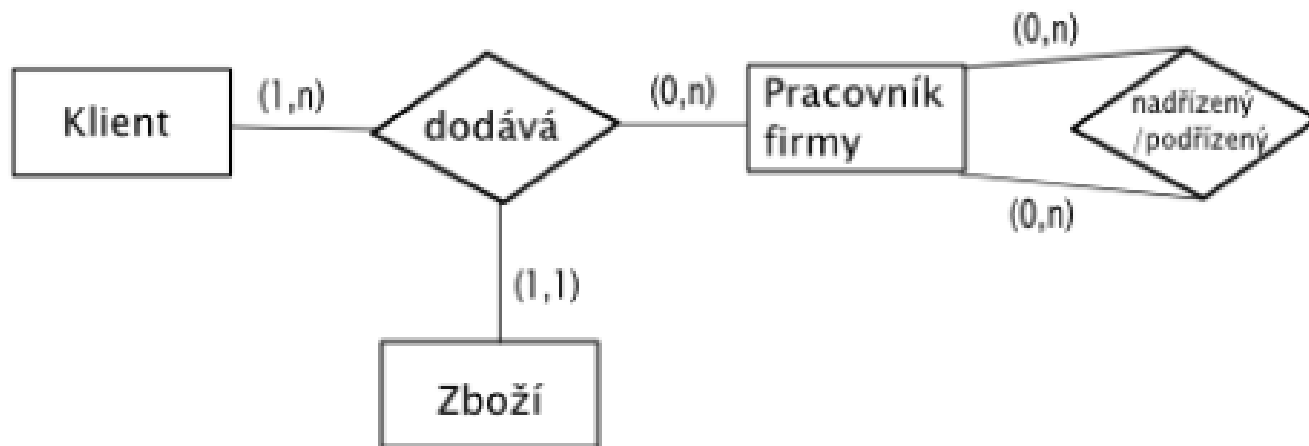
Vztahový typ

- 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 značena vedle čáry spojující vztahový typ s entitou.
- Entity ve vztahu mohou mít navíc povinné či nepovinné členství (vstupovat do něj vždy či někdy)

Vztahový typ

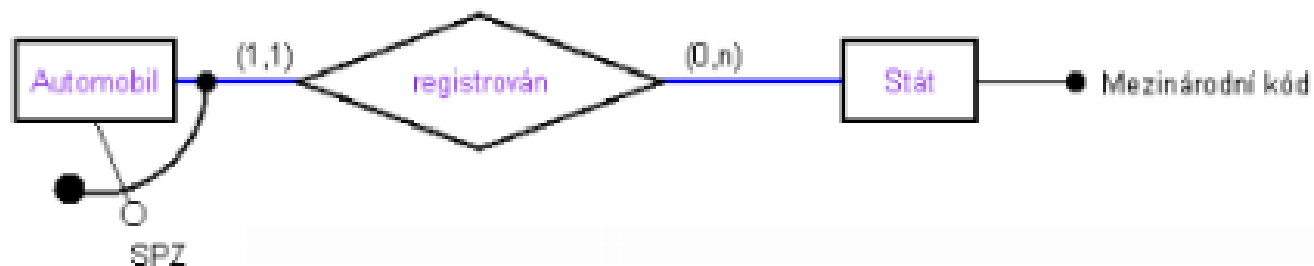
- Vztahy mohou být buď binární nebo **n-ární**, ale více než ternární vztahy se většinou neobjevují. Vztahy mohou být i **REKURZIVNÍ**, tj. do vztahů vstupují entity stejného typu.
- Instance vztahového typu je jednoznačně určena identifikátory instancí ve vztahu.
- Některé entitní typy mohou být spolu identifikovány (nebo přímo identifikovány) vztahem – pak se nazývají slabé entitní typy.

Vztahové typy



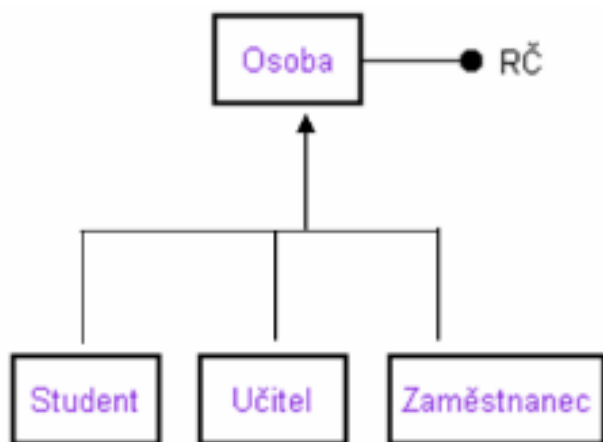
Obr. Ukazuje ternární vztah s různými kardinalitami – klientovi někdo dodává zboží jednou až n-krát, pracovník dodává nula až n-krát zboží (tj. jde o nepovinné členství ve vztahu, mohou existovat pracovníci, kteří nic nedodávají) a zboží je vždy dodáváno právě jednou. Na zaměstnancích je zároveň ukázán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

Slabý entitní typ



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

ISA hierarc



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)

Základní pojmy

- **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

Základní pojmy

- **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)

1. Určení typu entit

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

2. Určení typů relací

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

3. Určení atributů

- přiřazení jednotlivým entitám a vztahům
- např. Objednávka (číslo, datum, ...)

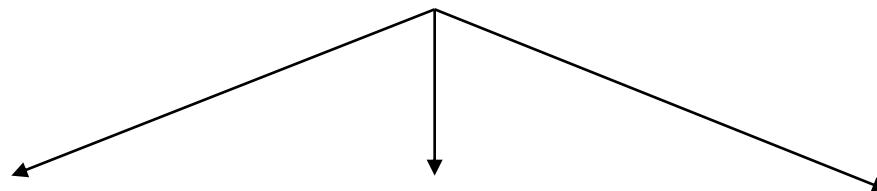
4. Určení integritních omezení

- zpřesnění navrženého modelu
- např. atribut datum je datového typu Datum a čas

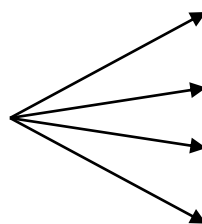
ER model

Relační tabulka

atributy



záznamy



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

FK **FK**

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

PK

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

PK **FK**

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

PK

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

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 <např. BeginTrans>.
- Transakce může skončit úspěšně nebo neúspěšně.
- Neúspěch může vzniknout: poruchou hardware, chybou programového vybavení, chybnými daty, apod.

Transakce

- Bod, od kterého lze považovat transakci z 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).

SQL

- 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 (skrze 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 TRANSAKcí

(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.

SQL - uplatnění

- 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 SŘBD v LAN - v síti s několika různými SRBD je SQL jedinou cestou pro jejich vzájemnou komunikaci.

Metodika, vývoj, typologie IS

Metodiky, metody, techniky, nástroje

- Metodika = souhrn etap, přístupů zásad.

Metodika 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

Metodiky, metody, techniky, nástroje

- 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.

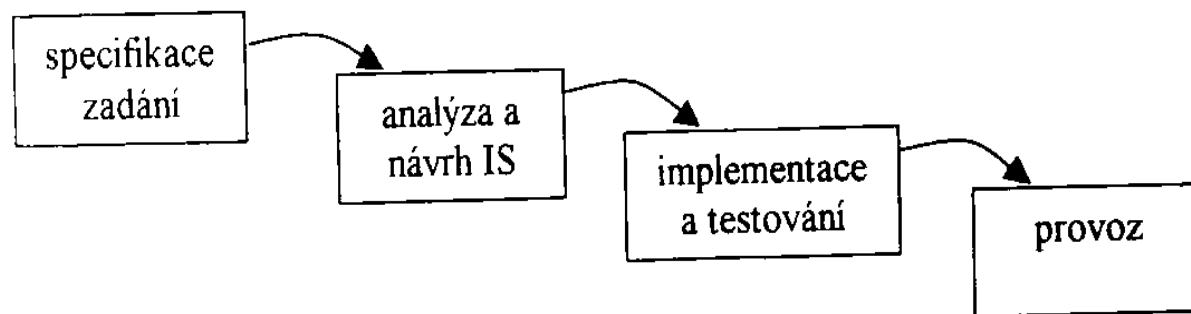
Metodiky, metody, techniky, nástroje

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

Vazby metodika-metoda-technika-nástroj

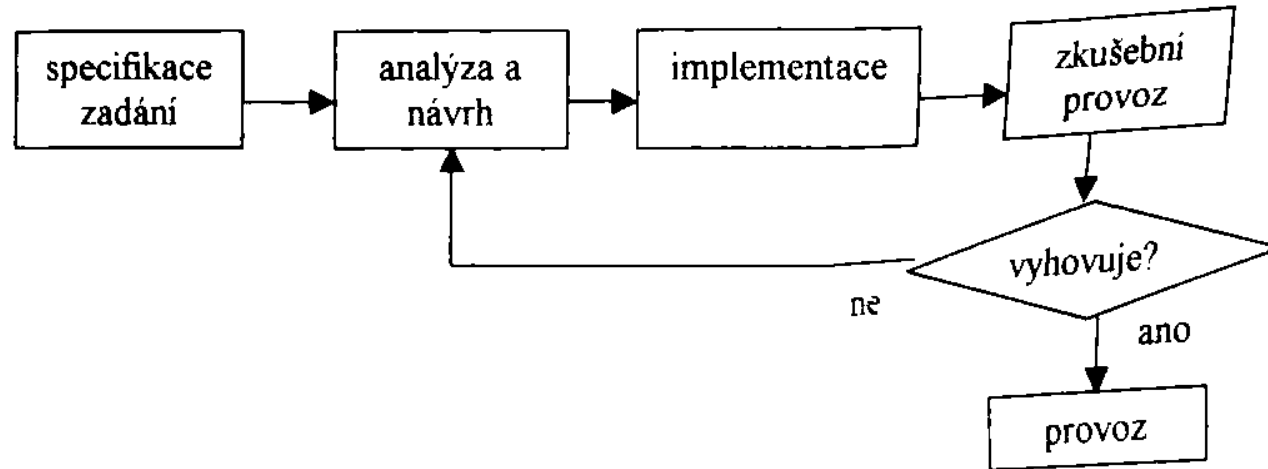
- Metodika doporučuje použití určitých metod v průběhu vývoje IS, metody pak využívají určitých technik a nástrojů. Není však možné prohlásit, že daná metoda patří jednoznačně k určité metodice. Některé metody jsou specificky využívány konkrétní metodikou. Většina metod je univerzálních, využívají různé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

Životní cyklus IS



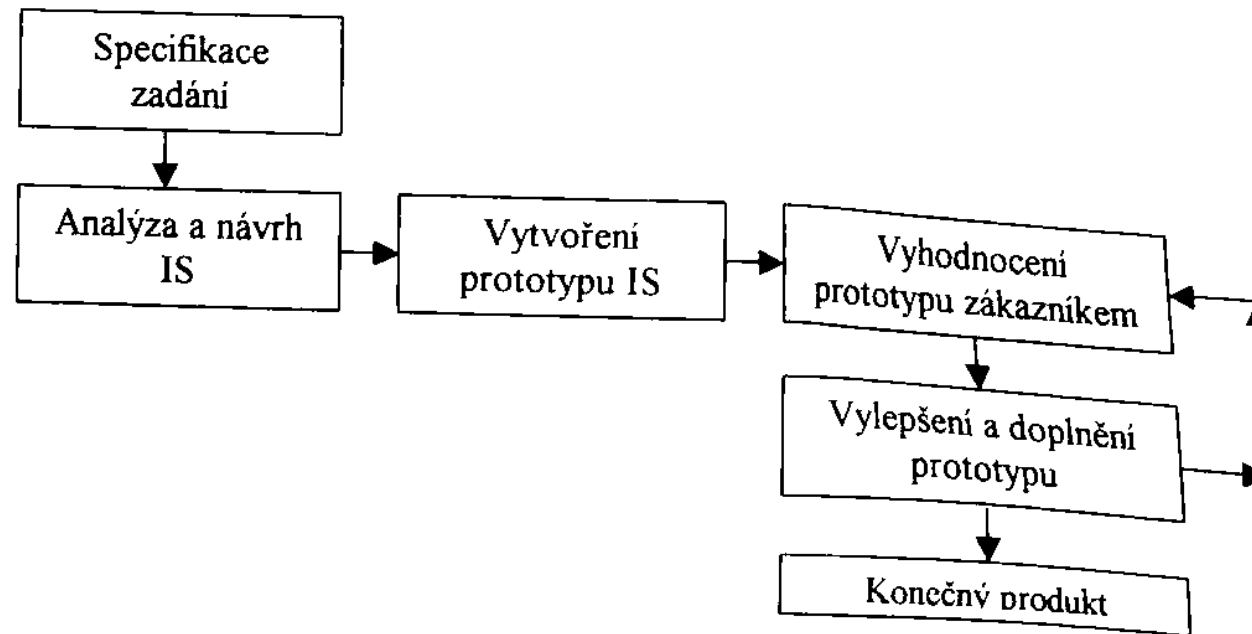
Obr 2.1 Model životního cyklu IS - "vodopád"

Životní cyklus IS



Obr 2.2 Model životního cyklu IS - "výzkumník"

Životní cyklus IS



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 (1)

- Hierarchický rozklad problematiky

rozdělení složitého systému na subsystemy a to až do potřebné úrovně podrobnosti.

Hierarchické rozdělení systému na subsystemy napomáhá plánovat, organizovat a kontrolovat práci vývojového týmu.

Základní prostředky pro boj se složitostí vývoje IS (2)

- 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.

Základní prostředky pro boj se složitostí vývoje IS (3)

- Modelování a srovnávání modelů
základní technika používaná během vývoje IS.

Základní prostředky pro boj se složitostí vývoje IS (4)

- 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

- Princip ABSTRAKCE

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

opakem

- KONKRETIZACE

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

Analýza, návrh IS

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)

Analýza, návrh IS

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.

Analýza, návrh IS

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.

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.

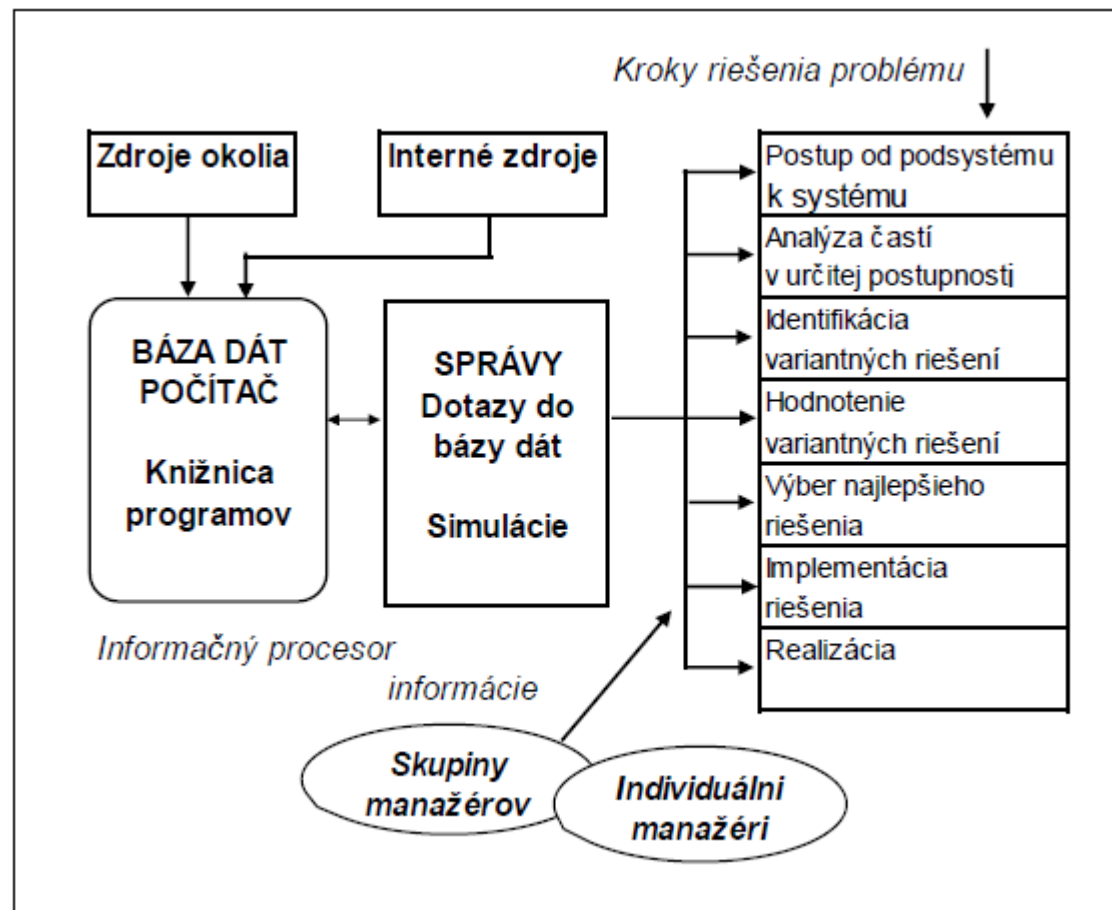
Typologie IS (pokračování)

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.

Obr. DSS



Typologie IS (pokračování)

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.

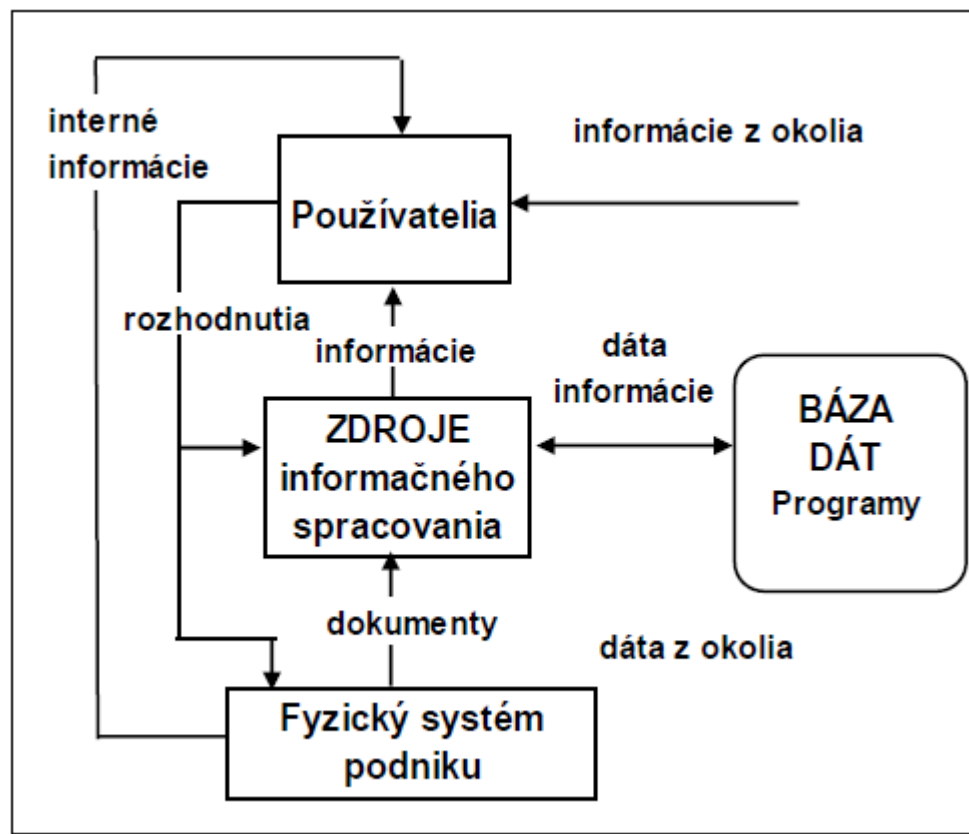
Typologie IS (pokračování)

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. MIS



Typologie IS (pokračování)

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.

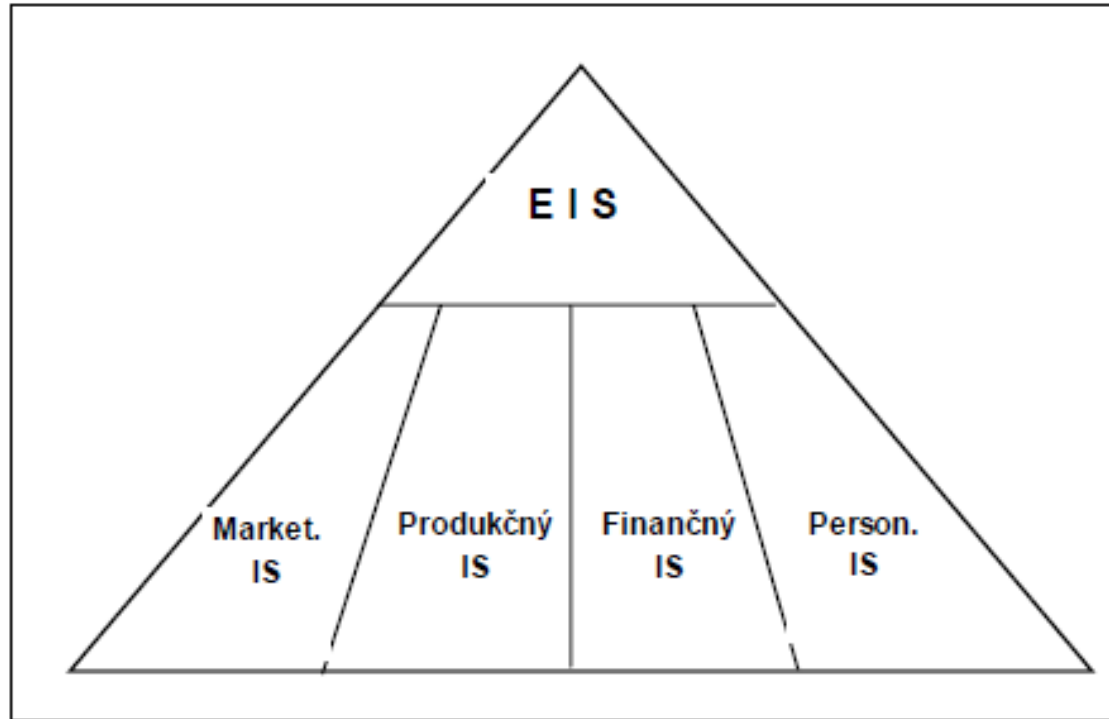
Integrovaním přímého řízení procesů s organizací výroby, zásobování a expedice vznikají integrované výrobní informační systémy (Computer Integrated Manufacturing – CIM).

Typologie IS (pokračování)

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.

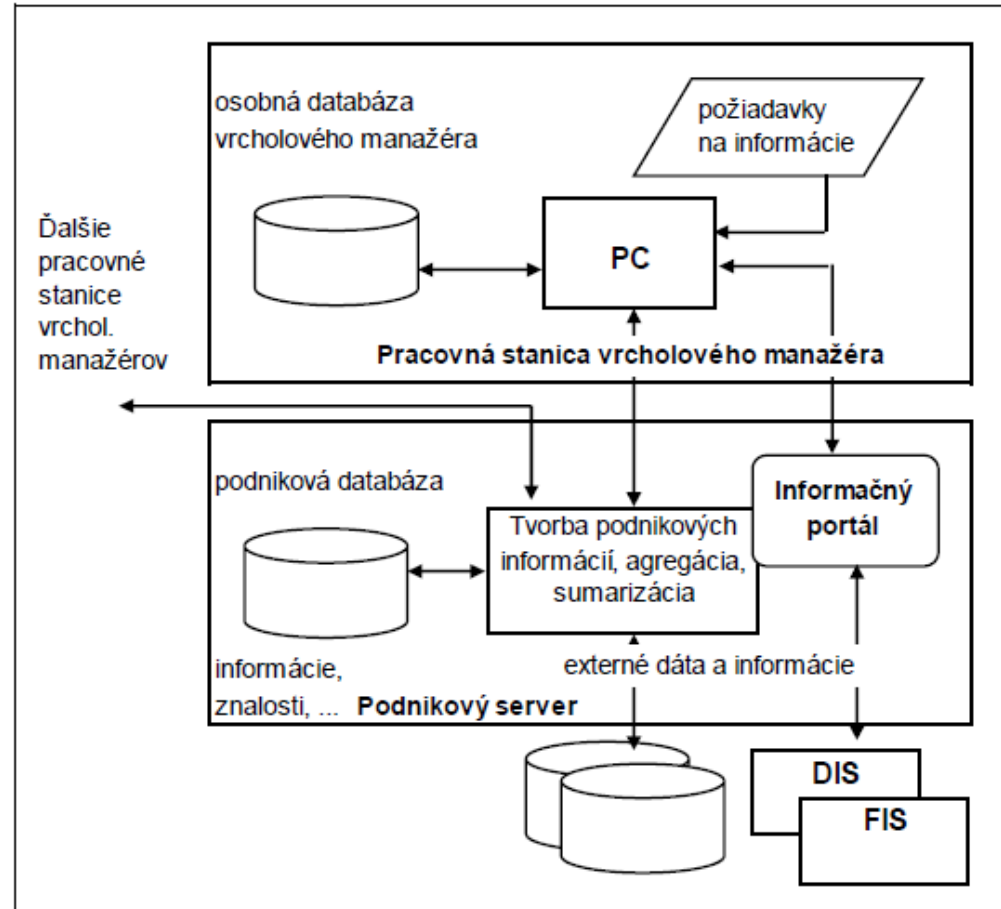
Obr. EIS



Srovnání MIS & DSS

<i>Kritérium</i>	<i>MIS</i>	<i>DSS</i>
Hľadisko podpory	organizácia	individuálne
Typ podpory	nepriama	priama
Podpora fáz riešenia problému	identifikácia, pochopenie a realizácia riešenia	všetky
Typy podporovaných problémov	všetky	semištruktúrované
Dôraz na	informácie	rozhodnutia

Obr. EIS a jeho propojení na DIS a FIS přes IS



Podpůrné IS

Kancelářské IS (Office Automation – OA)

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

Podpůrné IS

Útvarové systémy (Departmental Systems – DS)

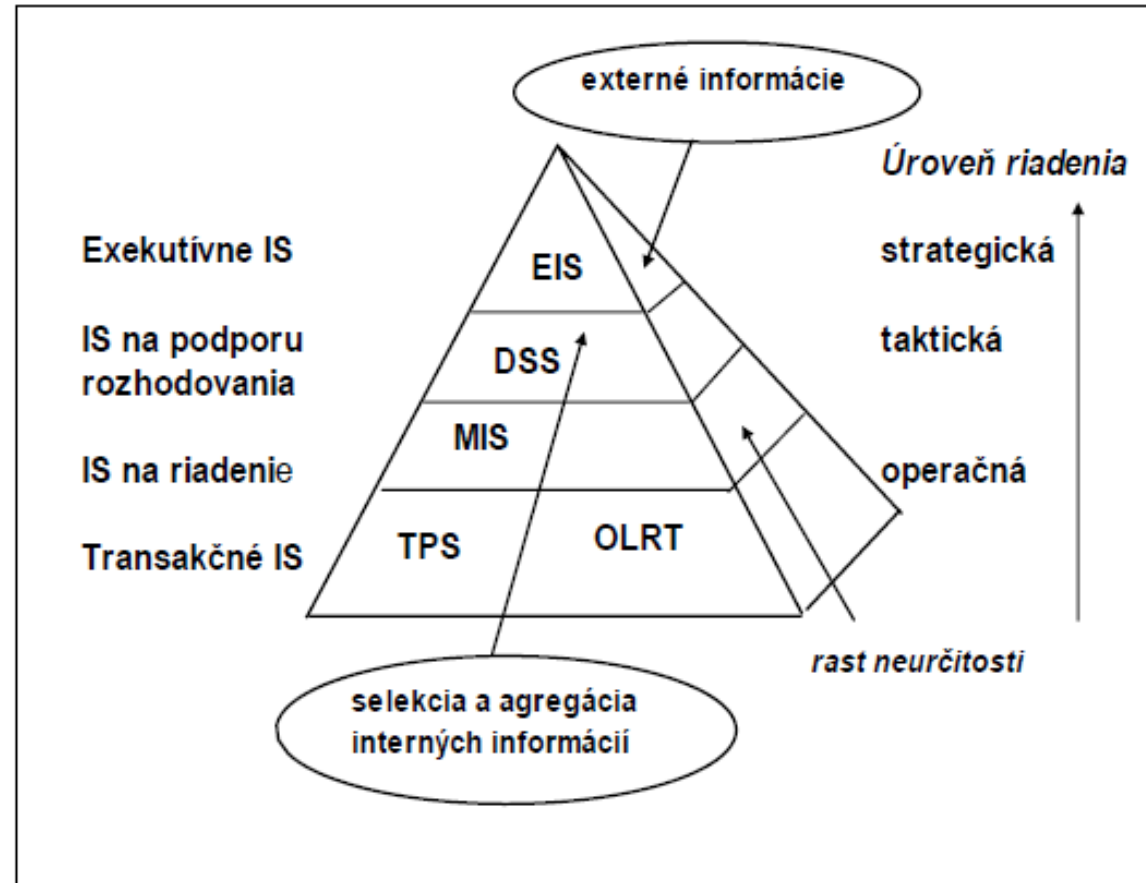
Jsou často spojením TPS, DSS a OA, ale jejich rozsah je redukován na určitý útvar nebo místo.

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



Modelování IS – Strukturovaný a objektově orientovaný přístup (UML)

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)

Princip tří architektur

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 subsystemy a vazbami mezi nimi.

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

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

Princip tří architektur

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).

Princip tří architektur

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 MODELy 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 subsystémy 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 subsystémy)

- **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 ORIENTO VANÝ 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.

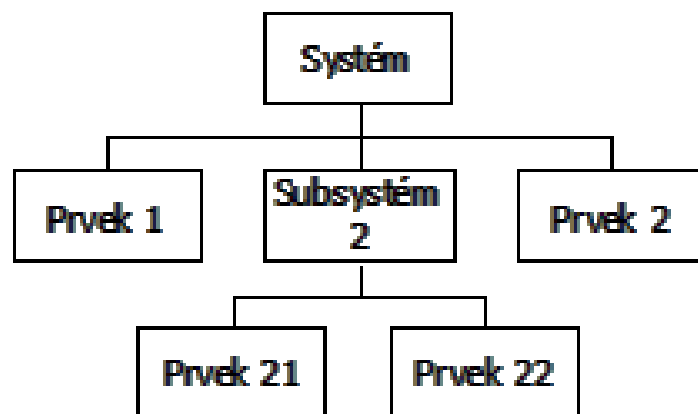
MODELY

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 subsystémy a prvky pomocí stromových diagramů.



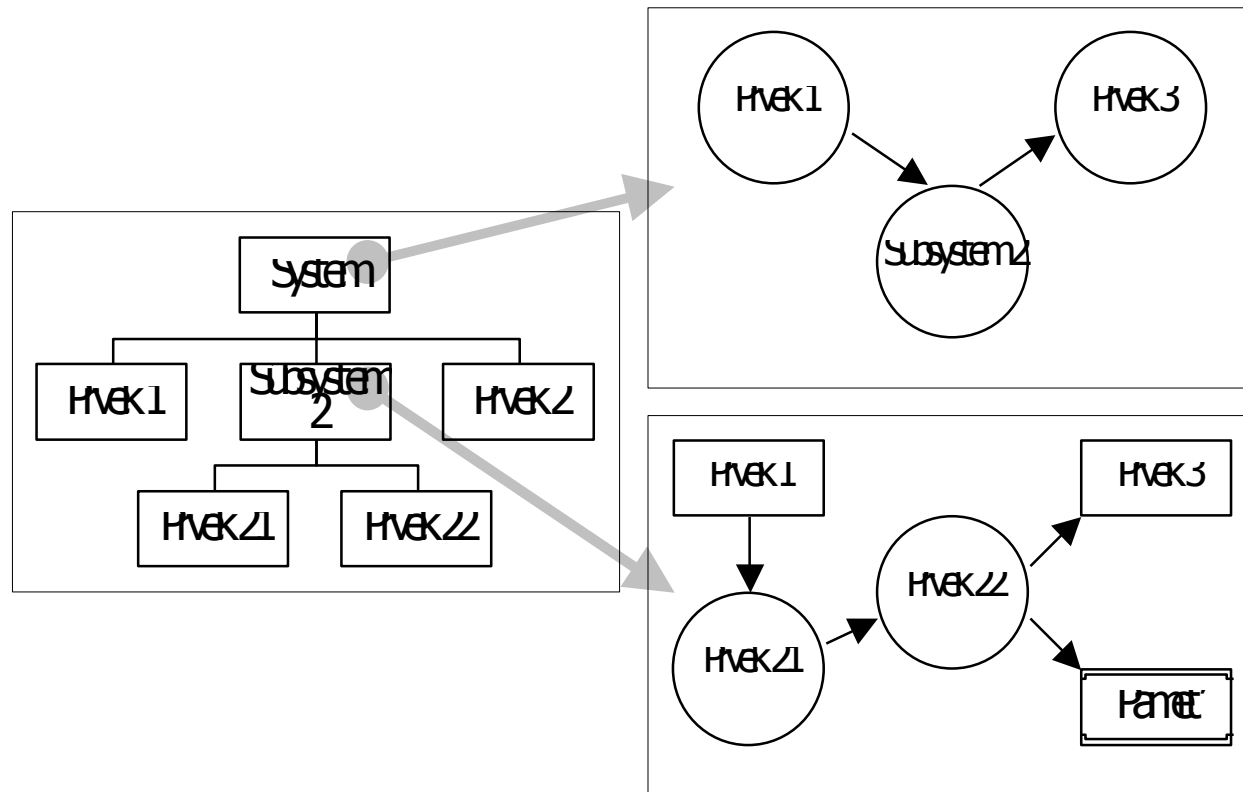
Informační toky

- 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.

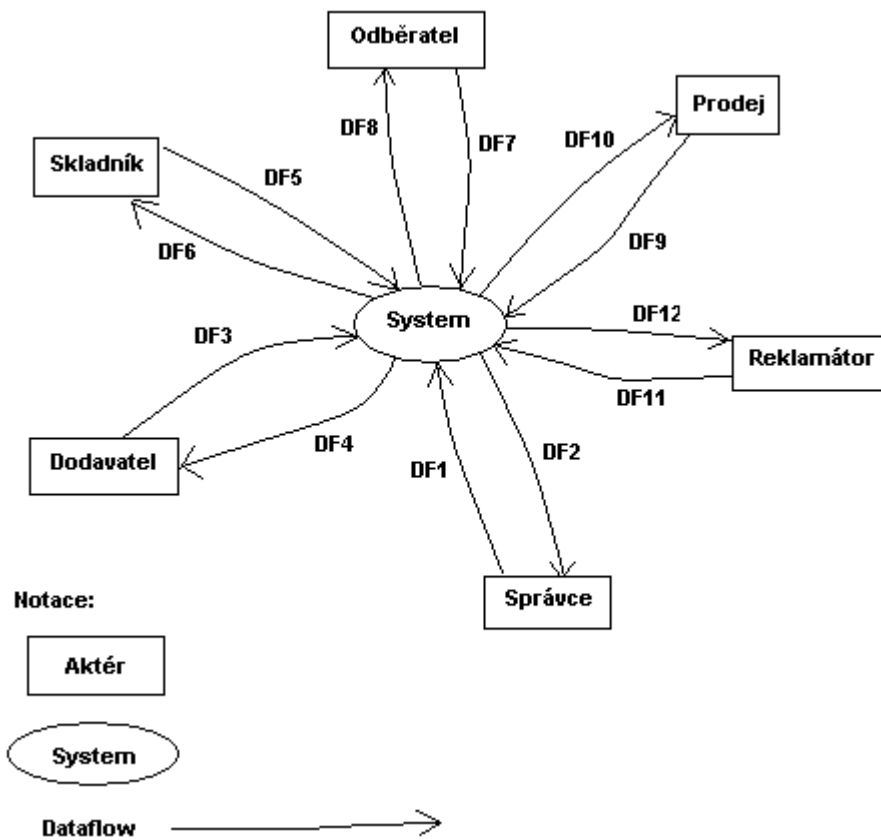
Informační toky

- **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 subsystemy 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.

Informační toky



Kontextový diagram



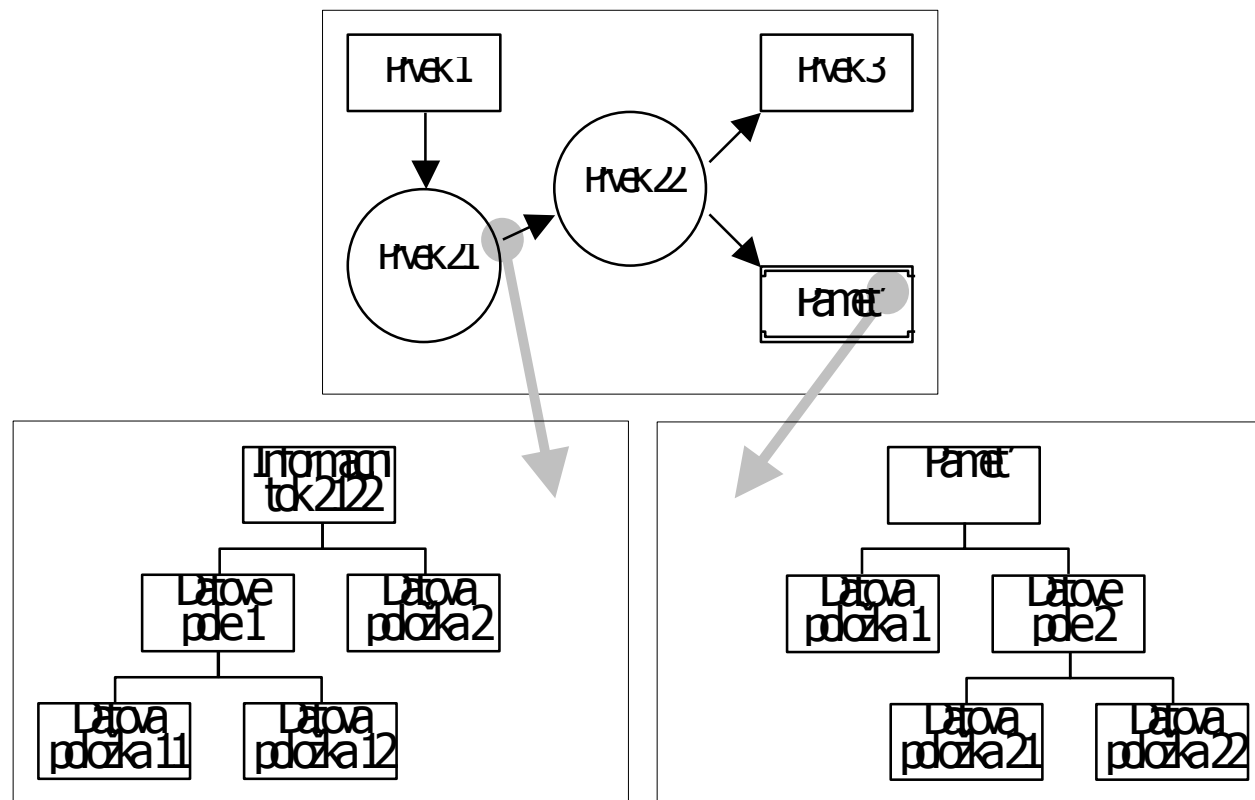
Datové struktury

- 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.**

Datové struktury

- 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.

Datové struktury

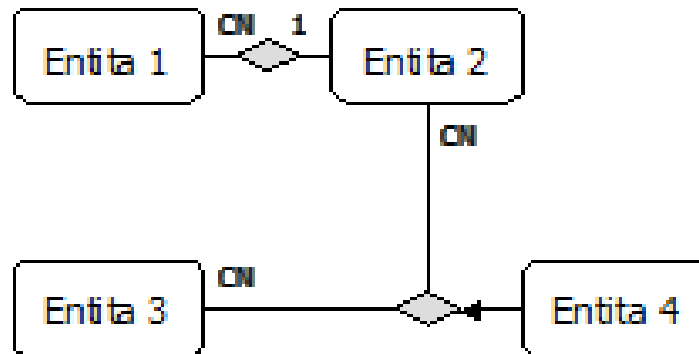


ER model

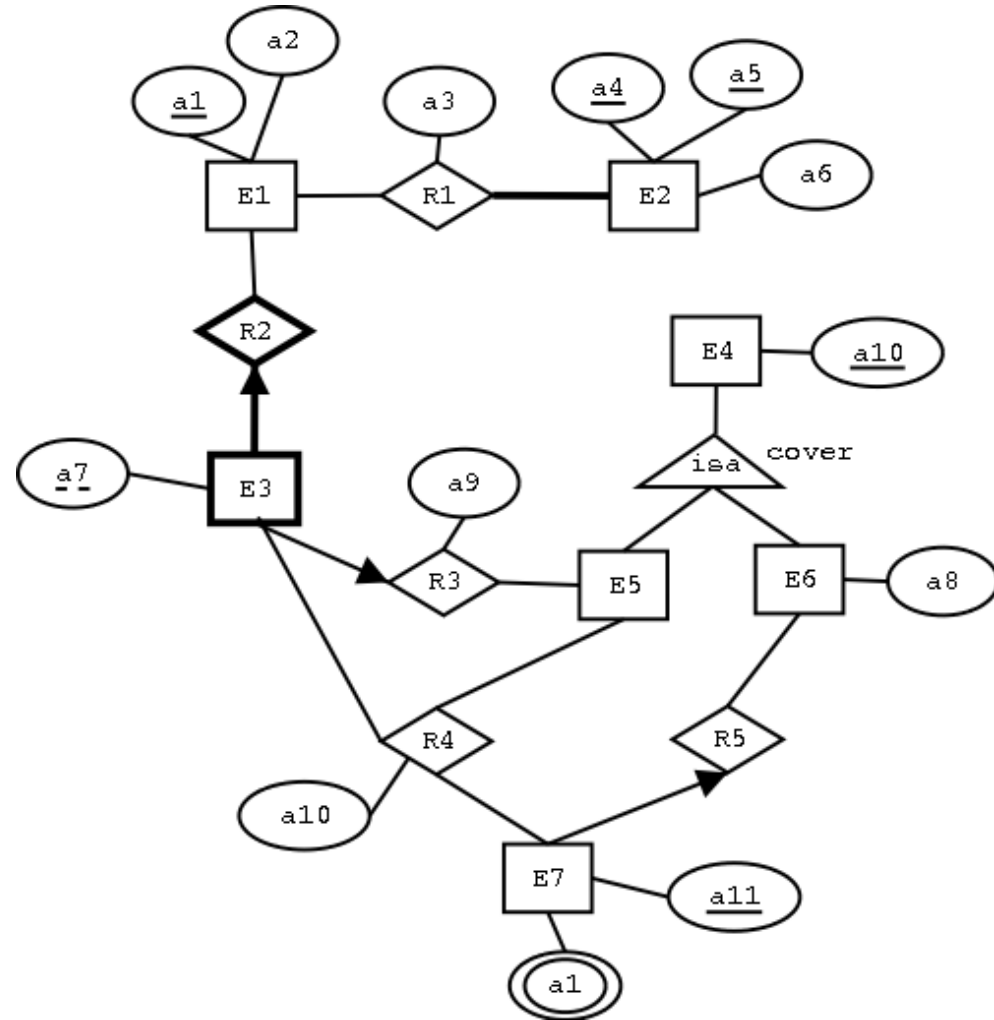
- ER model – model entit a jejich vzájemných vztahů.
- Vhodný pro analýzu systému v případě, kdy složitost systému spočívá spíše ve složitosti struktury dat než ve složitosti jeho funkčních složek.
- ER model zachycuje formou síťového grafu objekty reálného světa a vztahy mezi nimi. Množiny objektů reálného 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.

ER model

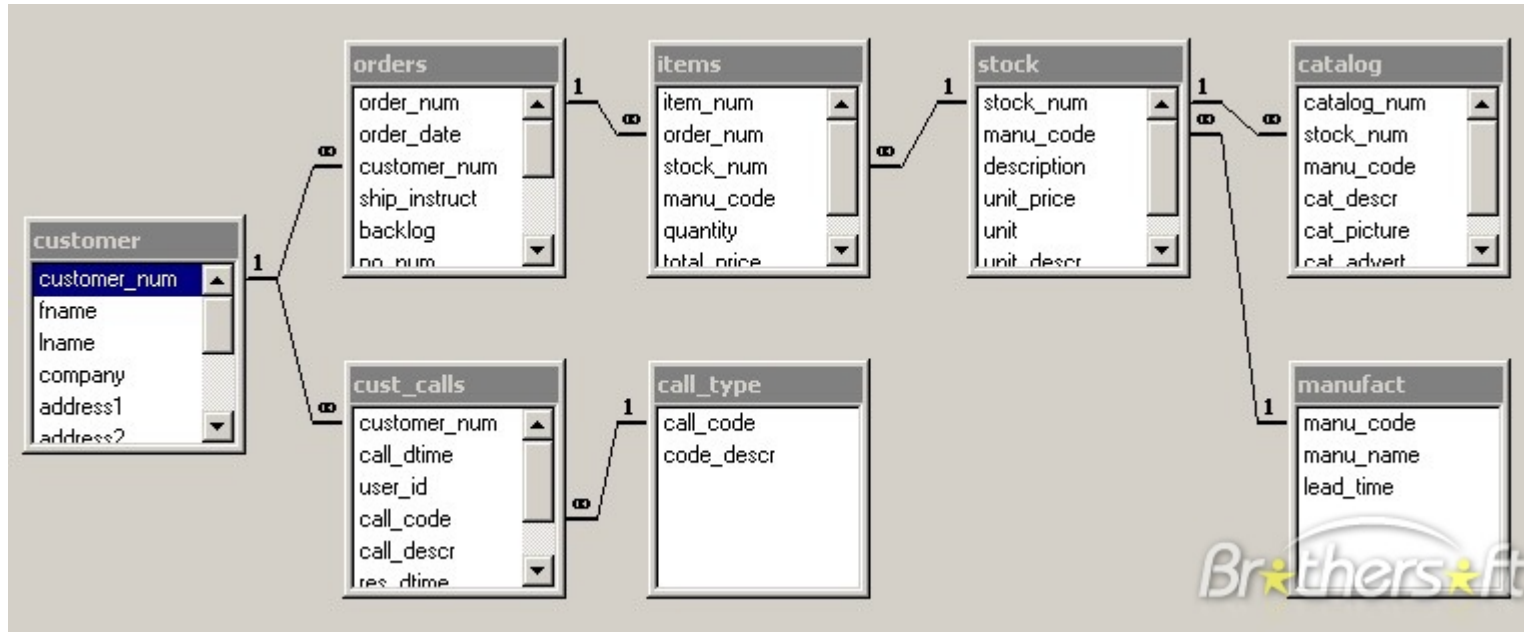
- Relace mezi entitami jsou specifikovány kardinalitou a těsností vazby. ER modely se využívají pro tvorbu modelů dat na logické neboli konceptuální úrovni, tedy modelů dat nezávislých na jejich fyzické realizaci prostřednictvím specifického databázového systému.



ER model



ER model



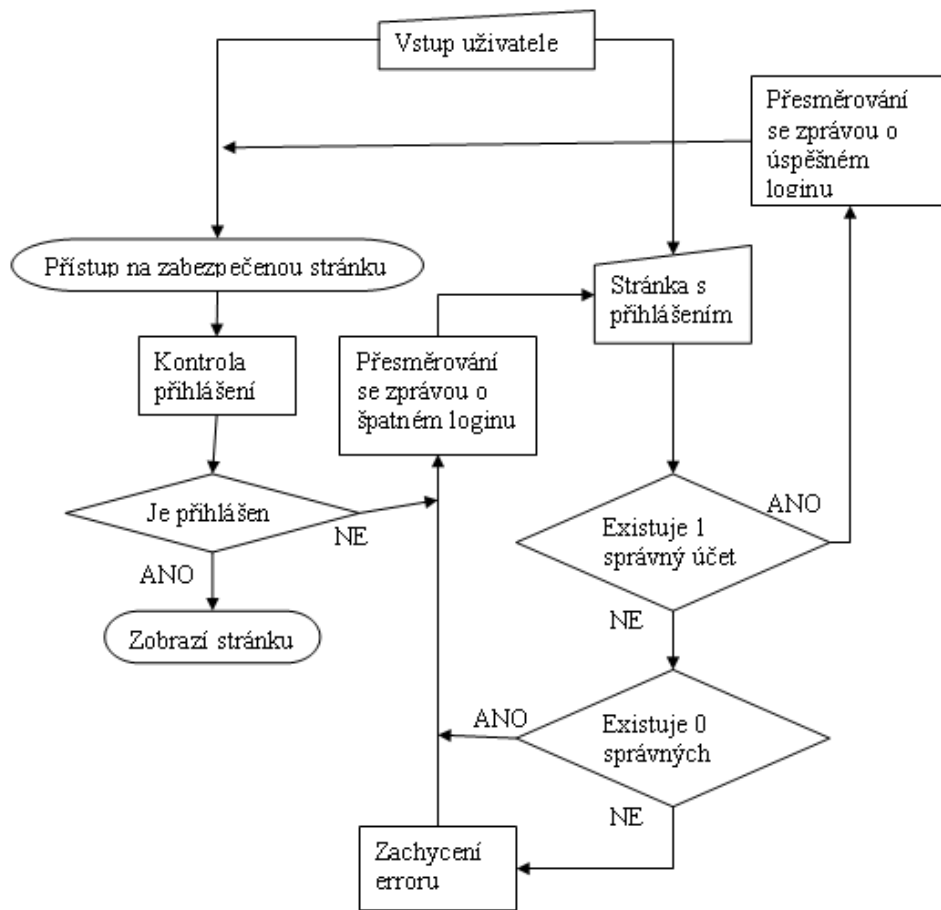
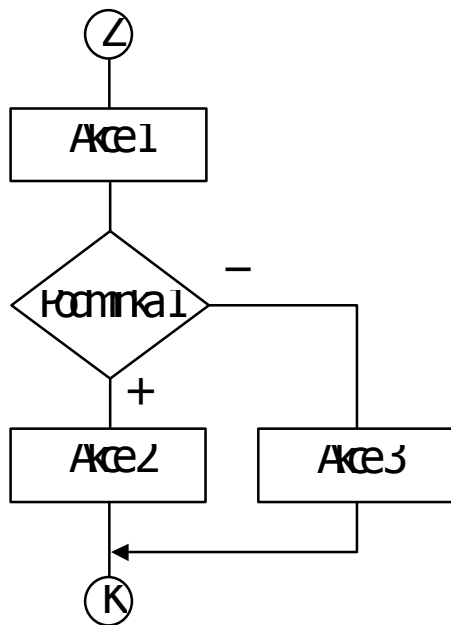
Metody popisu chování

- 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.

Metody popisu chování

- 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.

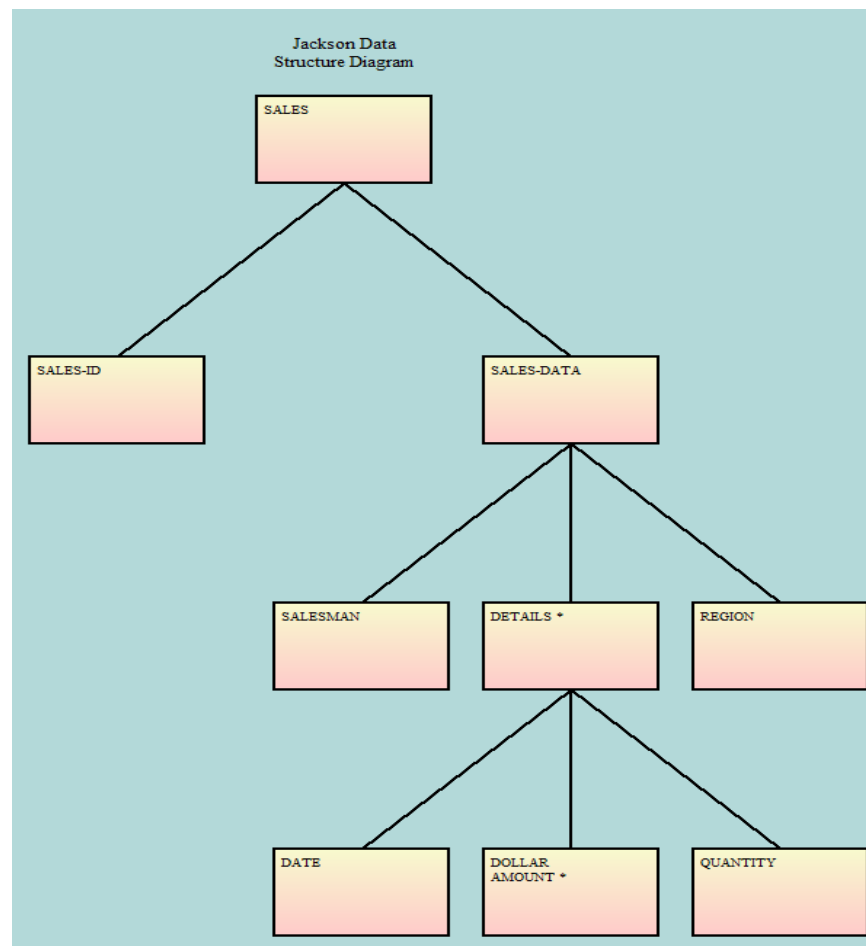
Vývojový diagram



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.

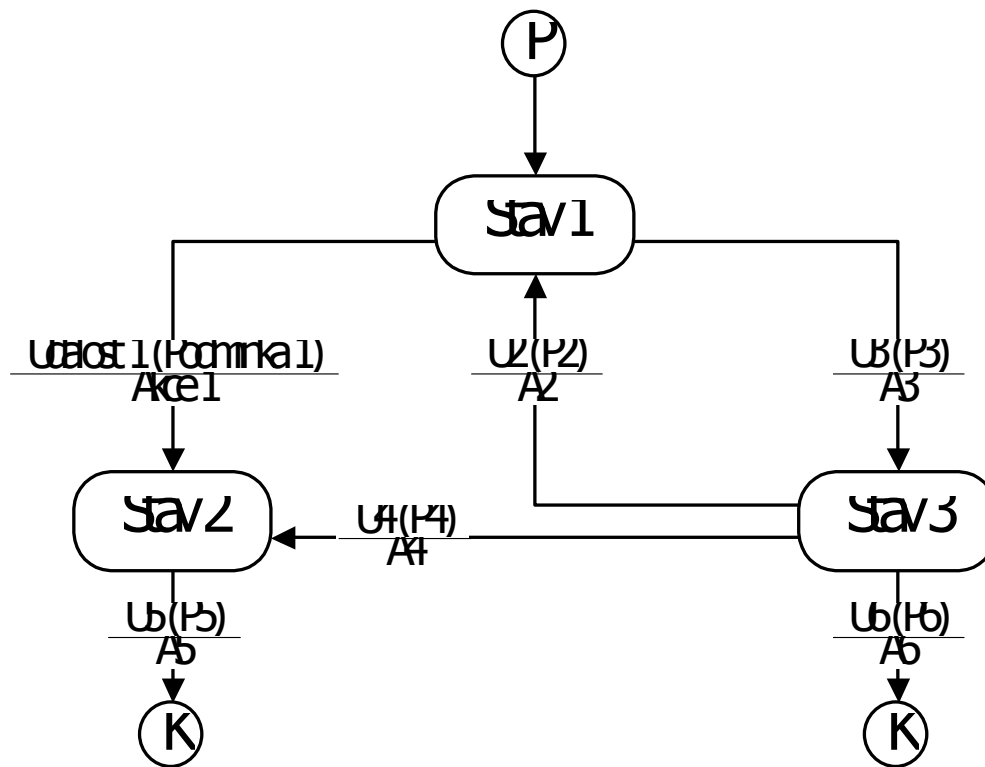
Jacksonův diagram



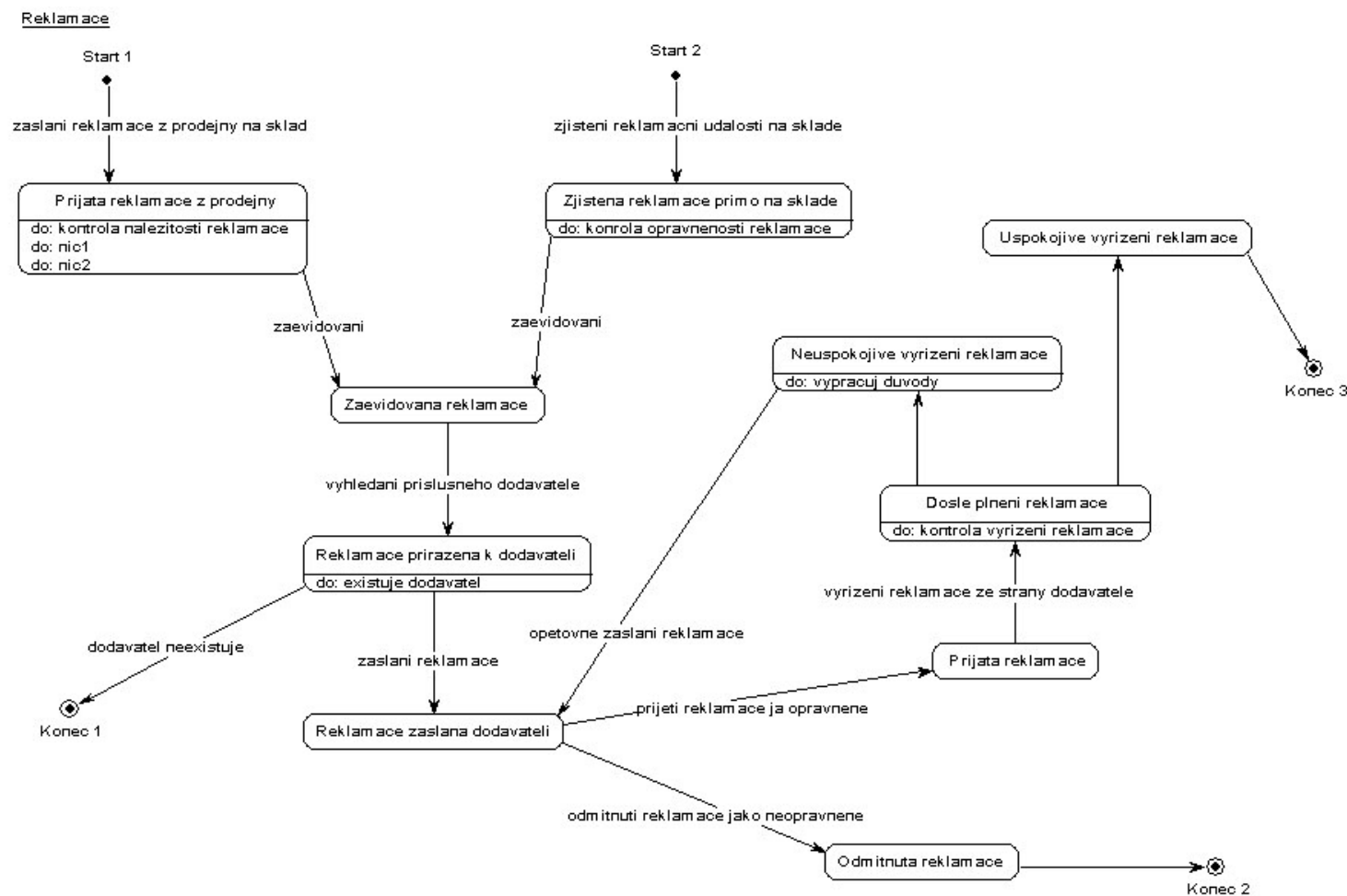
Metoda stavových diagramů

- grafická metoda popisu chování použitelná pro vyjádření jednodušších algoritmů nebo pro popis algoritmů na hrubé úrovni.
- Systém je popsán SÍŤOVÝM GRAFEM, jehož **uzly znázorňují stav systému a hrany naznačují možné přechody mezi stavy** s vyjádřením podmínek přechodu a akcí s přechodem spojených.
- Akce a stavy systému pouze symbolicky označují funkce a jejich účinky. Algoritmickou náplň akcí je však nutné popsat s využitím jiných prostředků.

Stavový diagram



Stavový diagram



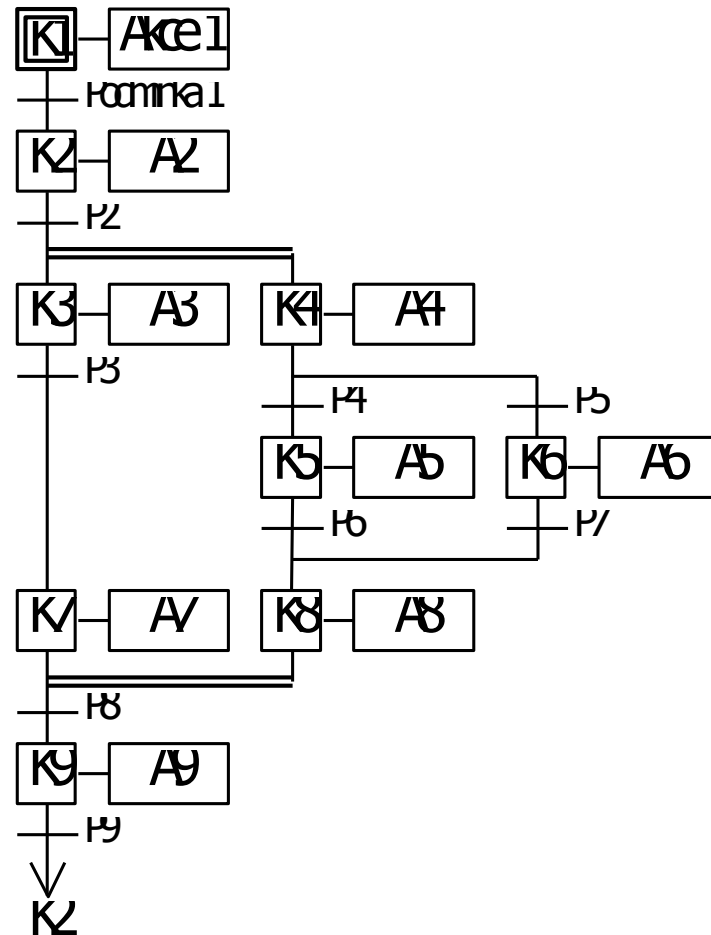
Metoda sekvenčních funkčních grafů

- Metoda sekvenčních funkčních grafů představuje grafickou metodu popisu chování systému vyhovující nejobecnějším nárokům na popis algoritmu. Základy metody jsou postaveny na principech Petriho sítí.
- Metoda funkčních kroků a přechodů byla poprvé uveřejněna v roce 1977 pod názvem GRAFCET (Graphe Fonctionnel de Commande Etape-Transition). V současné době se však s touto metodou lze setkat častěji pod názvem SFC (Sequential Function Chart).

Metoda sekvenčních funkčních grafů

- Metoda sekvenčních funkčních grafů využívá k popisu systému obdobně jako metoda stavových diagramů síťový graf.
- Zásadní rozdíl však spočívá v tom, že uzlem síťového grafu není stav systému, ale krok algoritmu, respektive akce s daným krokem spojená. Přejít z kroku do kroku je vázán podmínkou přechodu a aktivitou kroků této podmínce bezprostředně předcházejících. Síťový graf tedy zachycuje základní strukturu algoritmu, detailní popis je pak obsahem akcí spojených s dílčími kroky.

Sekvenční funkční graf

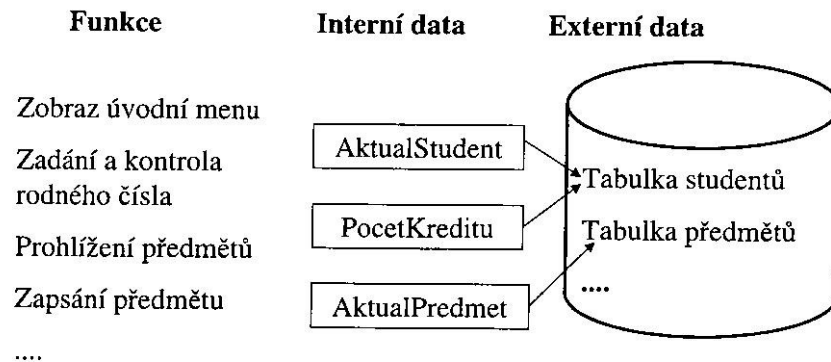


Objektově orientovaný přístup

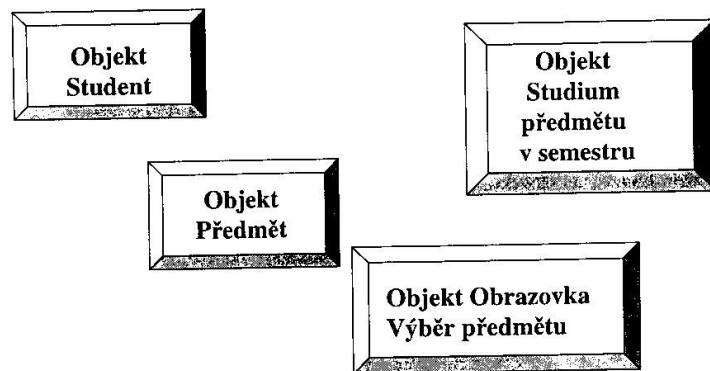
- Historicky mladší technika navazující na strukturovaný přístup.
- Přístup je založen na OBJEKTECH, jakožto strukturách, které mají definované vlastnosti (ATRIBUTY) a své chování (operace, které daný objekt může provádět).
- Vlastnosti i operace jsou „zapouzdřené“ v jednotlivých objektech.
- IS je chápán jako MNOŽINA spolupracujících OBJEKTŮ.
- Každý OBJEKT je schopen reagovat na události, které na něj působí jako IMPULS.

Objektově orientovaný přístup

Procedurální přístup



Objektový přístup



Objektově orientovaný přístup

- umožňuje lepší využití kódu než knihovny procedur.
- možnost znovupoužití.
- lze využít prostředky pro okenní rozhraní systémů.
- dávkové úlohy jsou interaktivní (objekt reaguje na více podnětů)

Objekt

Struktura objektu:

- 1) VNITŘNÍ PAMĚŤ
- 2) METODY OBJEKTU
- 3) JINÉ OBJEKTY
- 4) SCHOPNOST PŘIJMOUT A ZPRACOVAT ZPRÁVU Z VNĚJŠKU

Objektově orientované modelování

- způsob nazírání na IS pomocí abstrakce reálného světa. Základním stavebním prvkem je objekt (entita), která v sobě zahrnuje jak datovou strukturu popisující určitou entitu, tak i pravidla chování této entity.

Vizuální modelování (OOM)

Vlastnosti

- “Informace v obrázcích”
- Prostředek komunikace (pohledy, modely, diagramy)
- Zachycuje obchodní procesy, věcnou problematiku, architekturu systému
- Je podporováno silnou notací a objektovými metodologiemi

Metodologie

- **Booch**, Coad,Yourdon
- OMT (Object Modeling Technik), **Rumbaugh**
- OOSE (Object-Oriented Software Technik)
- **Jacobson** (scénáře)
- EFEM (Extrémní efektivní modelování)

Vizuální modelování (OOM)

- Modely vytváříme pro pochopení komplexnosti systému jako celku.

Za pomocí modelů lze:

- vizualizovat stávající nebo vytvářený systém,
- specifikovat strukturu nebo chování systému,
- získat základ pro konstrukci systému,
- dokumentovat rozhodnutí o systému.

Objekty

Charakteristické vlastnosti objektů:

- **Jedinečnost** – znamená, že každý objekt je rozlišitelnou entitou, i když má jinak stejné kvalitativní i kvantitativní charakteristiky. K rozlišení se používá identifikační číslo.
- **Zatřiditelnost** – objekty se stejnou datovou strukturou (atributy) a chováním (operacemi) jsou seskupovány do tříd.
- **Mnohotvárnost** – znamená, že stejně označená operace se může chovat rozdílně a dávat rozdílné výsledky pro různé třídy objektů.
- **Dědičnost** – je sdílení atributů a operací mezi třídami založenými na hierarchických vztazích. Třída může být definována poměrně široce a potom ji lze následně zjemňovat do podtříd. Každá podtřída pak zahrnuje vlastnosti své nadřazené třídy.

Objektově orientované modelování

- Založeno na principech, které vyjadřují podstatu objektového přístupu.

4 principy OOP:

- **Zobecnění** – představuje zaměření na podstatné vnitřní aspekty objektů a ignoruje jeho nepodstatné vedlejší vlastnosti. To znamená, že klade důraz na to, co objekt je, a co by měl dělat, a ne na to, jak by to měl dělat.
- **Zapouzdření** – znamená oddělení externích aspektů objektu, které jsou dostupné ostatním objektům, od interních implementačních detailů objektu, zakrytých ostatním objektům. To znamená, že implementace objektů může být změněna, aniž by se to dotklo aplikací, ze kterých je objekt použit.
- **Sdílení** – je umožněno dědičností datových struktur a chováním mezi několika podobnými třídami bez redundance. To umožňuje opakované použití již navržených a ověřených programů v dalších aplikacích.
- **Spolupůsobení** – jednoznačnost, zatřiditelnost, mnohotvárnost a dědičnost charakterizují hlavní proud objektově orientovaných jazyků. Každý z těchto aspektů může být izolován, ale dohromady působí symetricky.

Výhody a nevýhody OOP a tradiční analýzy

Výhodou tradičních popisů IS oproti OOP bylo, že jejich osvojení bylo relativně snadnější, neboť jejich rozsah od zadání softwarové úlohy až po bezprostřední zadání programových dat a procedur byl řešen nezávisle na pochopení funkcí reálného podnikového systému jako celku a jeho potřeb nejen informačních.

Nevýhody objektového přístupu

- Zavedení nového vyjadřovacího prostředku a nového způsobu myšlení, což je dosti náročné pro lidi, zúčastněné na procesu modelování, a později pak i na programování.
- Používání počítačové podpory pomocí vhodného CASE, bez níž nelze provést kvalitní analýzu i relativně malého systému. To opět vyžaduje zvládnutí práci s daným CASE (Computer Aided System Engineering).
- Technologickou kázeň s tím nutně spojenou.

Výhody OOP

- Možnost vysledovat na modelu vnitřní vztahy prvků v podnikovém systému a vytvářet software pro „poskytovatele služeb“, tj. pro prvky podniku, které poskytují své služby jiným podnikovým prvkům.
- Takto vytvářený software může být opakovatelný, zejména při použití knihoven standardizovaných tříd.
- Možnost vytvářet software podstatně pružnější vzhledem k proměnlivým potřebám podniku. Z hlediska programů je to dáno tím, že dříve byly strukturované programy vytvářeny podle pevné struktury potřeb podniku v době jejich vzniku, bez snadné možnosti změn.

Historie OOP

- O OOP se začíná hovořit v 2. pol. 80. let 20. stol.
- Na počátku 90. let již existovala řada metod, nástrojů a technik založených na OOP.
- Snaha o integraci jednotlivých metod a o unifikaci OOP → Grady Booch & James Rumbaugh publikovali v r. 95 metodu s názvem Unified Method (vycházela z metod Booch a OMT).

Historie OOP

- V r. 95 se k autorům Unified Method Jacobson (autor metody Objectory/OOSE).
- Vznikají další verze Unified Method.
- Ukazuje se, že se jen těžko podaří prosadit jednotné postupy a doporučení a ucelenou OO metodu vývoje IS.

Historie OOP - UML

- Přejmenováním Unified Method na Unified Modeling Language (UML) vznikl nástroj – grafický jazyk pro tvorbu modelů IS použitím OOP.
- UML byl přijat sdružením OMG (Object Management Group – sdružení usilující o unifikovaný rozvoj OOp) jako doporučený standard notace pro tvorbu modelů IS.

UML

- V současnosti je UML nejrozšířenější objektovou notací.
- Zahrnuje několik druhů modelů (diagramů) IS a grafických vyjadřovacích prostředků pro jejich konstrukci.
- UML dnes podporují všechny CASE nástroje pro objektovou analýzu a návrh IS, OOM začleňují UML mezi své výrazové prostředky.

UML

V současnosti UML poskytuje:

- pravidla pro pojmenování, rozsah platnosti, rozsah viditelnosti, omezení, prezentaci modelu,
- různé specifikace,
- rozšiřitelnost jako jsou stereotypy, dodané hodnoty.

UML

Definice:

UML je standardní jazyk pro vizualizaci, specifikaci, konstrukci a dokumentaci prvků projektu, ve kterém hraje významnou roli vývoj software.

Stavební kameny:

- artefakty (prvky),
- vztahy,
- diagramy.

UML – skupiny ARTEFAKTŮ

Týkající se struktury systému

- tvoří statickou část modelu,
- třídy, rozhraní, use case, komponenta.

Týkající se chování systému

- tvoří dynamickou část modelu,
- interakce, stavy, aktivity.

Týkající se organizace systému

- package.

Týkající se vysvětlení účelu

- popis, anotace, poznámka.

UML - VZTAHY

Agregace (závislost)

- jeden prvek závisí na druhém.

Asociace

- propojení prvků.

Dědičnost

- specializace/generalizace

UML - DIAGRAMY

Grafická reprezentace obsahu modelu

- zachycení prvků a jejich vztahů.

Pohled na systém z různých perspektiv.

Různé typy diagramů

- Diagram užití (use case), tříd, objektů, sekvenční, spolupráce, stavový, aktivit, komponent, nasazení, balíčků (package).

UML - DIAGRAMY

1. Diagram Use case
2. Sekvenční diagram
3. Diagram spolupráce
4. Diagram tříd
5. Stavový diagram
6. Diagram aktivit
7. Diagram nasazení

Diagram Use case

- jeho vypracování je obsahem use case analýzy, zachycuje funkcionalitu systému z pohledu uživatele, popisuje chování systému z hlediska uživatele.

Prvky diagramu use case:

Aktor (vymezením aktorů specifikujeme okolí systému a vymezíme jeho hranice).

Use case (případ užití, typ jednání, funkcionalita, systému, kterou využívá aktor),

Vztahy (mezi aktorem a use case, mezi use case, výjimečně mezi aktory).

Diagram Use case

Aktor = kdokoliv nebo cokoliv mimo systém, kdo nějak komunikuje a interaguje se systémem.

- zachycení okolí systému,
- prvky aktivně komunikující se systémem:
 - uživatelé
 - jiné softwarové systémy
 - čas

Diagram Use case

Use case (typ jednání, případ užití)

jakákoliv funkčnost, která dává měřitelnou hodnotu uživatelům tohoto systému.

reprezentuje ucelenou funkcionalitu problémové domény

Diagram Use case

Typy vztahů v UCD:

- 1) Vztah mezi AKTOREM a USE CASE – aktor komunikuje se systémem (vyvolává a účastní se USE CASE)
- 2) Vztahy mezi USE CASE

Use case mohou vzájemně spolupracovat (slouží pro zjednodušení modelu a podobá se dekompozici).

Vztahy mezi USE CASE

Dva typy vztahů mezi use case:

- **Vazba include (dříve uses)**

povinný vztah, oba spojené use case se musí povinně provést, použijeme tam, kde se část use case v navrhovaném systému může opakovat.

- **Vazba extends**

rozšiřující vztah, za jistých podmínek se vykonávají oba use case, použijeme pro volitelné chování, pro chování za specifických podmínek, pro chování podle volby aktora.

Popis USE CASE - scénář

- Stručná charakteristika (1 - 3 věty)
- Scénář:
 - nutné podmínky před spuštěním,
 - nutné podmínky po ukončení,
 - tok událostí - sekvence akcí.
- Jeden scénář “HAPPY DAY” (obsahuje základní tok událostí a subtoky).
- Ostatní scénáře jsou alternativní, chybové.

Scénář (tok událostí) pro Use Case Př.: Evidence rezervace (půjčovna CD)

Předpoklady

Tento Use Case začne, když člen nemůže být uspokojen, protože dané CD není momentálně na skladě, nebo daný titul není v půjčovně k dispozici.

Hlavní tok

Tento Use Case začíná, když člen předloží asistentovi svoje identifikační číslo a název titulu, který si chce zarezervovat. Asistent zkontroluje existenci člena v databázi členů (**A-1**), zkontroluje, zda titul existuje v databázi titulů (**A-2**) a zkontroluje, zda jsou všechny kopie daného titulu zapůjčeny (**A-3**).

Pokud má někdo kopii zapůjčenu déle než 10 dní, je upomenut o navrácení (**S-1**). Je založen záznam o rezervaci této kopie pro daného člena. Je vytištěn doklad o rezervaci titulu.

Subtoky

S-1: Asistent vyhledá všechny členy, který mají půjčený daný titul a zkontroluje délku jejich půjčky. Pro ty, kteří mají půjčku delší než 10 dnů, vytiskne upomínku.

Alternativní toky

A-1 : Je vloženo špatné ID člena, nebo člen neexistuje. Asistent může opakovat vstup ID nebo vložit údaje o členu (bude řešeno v Use Case Přidání nového člena), nebo ukončit Use Case.

A-2 : Je vložen špatný titul, nebo titul neexistuje v půjčovně. Asistent ukončí Use Case (není založena rezervace) a vytiskne objednávku na daný titul (Use Case Objednání materiálu).

A-3 : Asistent zjistí kdo má půjčené kopie a vloží rezervaci pro člena.

Use case diagram

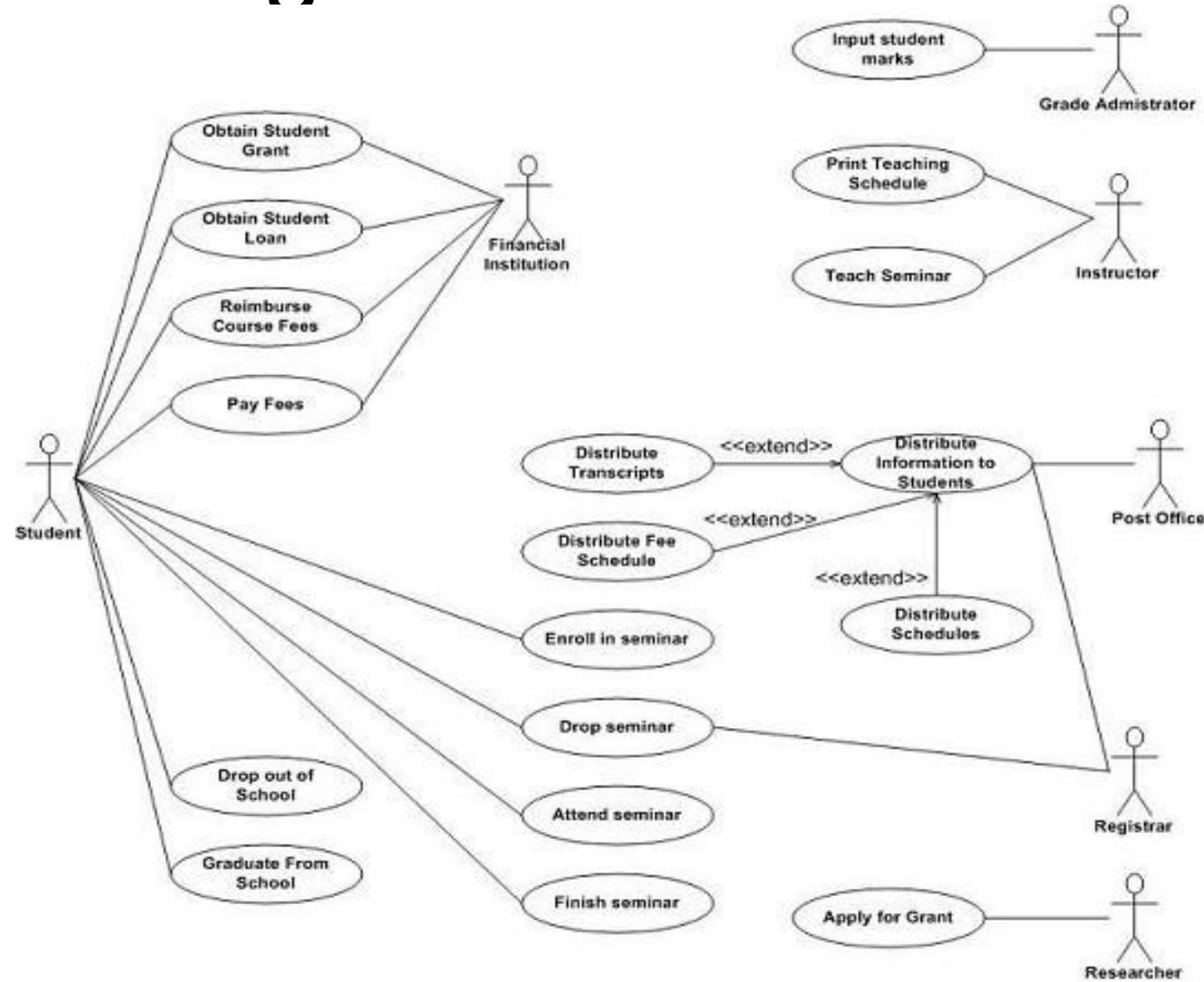
Informace o use case:

- jde o vizualizaci funkcionality, další informace musím spravovat v textové podobě,
- use case musí mít tyto náležitosti (jméno, popis – účel, kdo a co jej používá, související use case, hlavní a alternativní scénář, nepovinně poznámky).

Význam use case diagramu

- zachycení požadavků na systém,
- vizualizace a organizace požadavků ve standardní formě,
- pro nalezení objektů, tříd a zodpovědností z popisu scénářů.

Use case diagram



Sekvenční diagram UML

- Zachycuje interakci mezi objekty, zachycuje zasílání zpráv mezi objekty v rámci systému.
- Zachycuje dynamické chování s orientací na čas.

Vlastnosti sekvenčního diagramu:

- Objekty sekvenčního diagramu spolu komunikují pomocí zasílání zpráv.
- Popisuje jeden průchod zpráv systémem.
- Nemá přímé výrazové prostředky pro smyčky, větvení a podmínky.
- Pro jednoduché případy použijí poznámky.
- Složité případy řeším separátními diagramy.

Sekvenční diagram

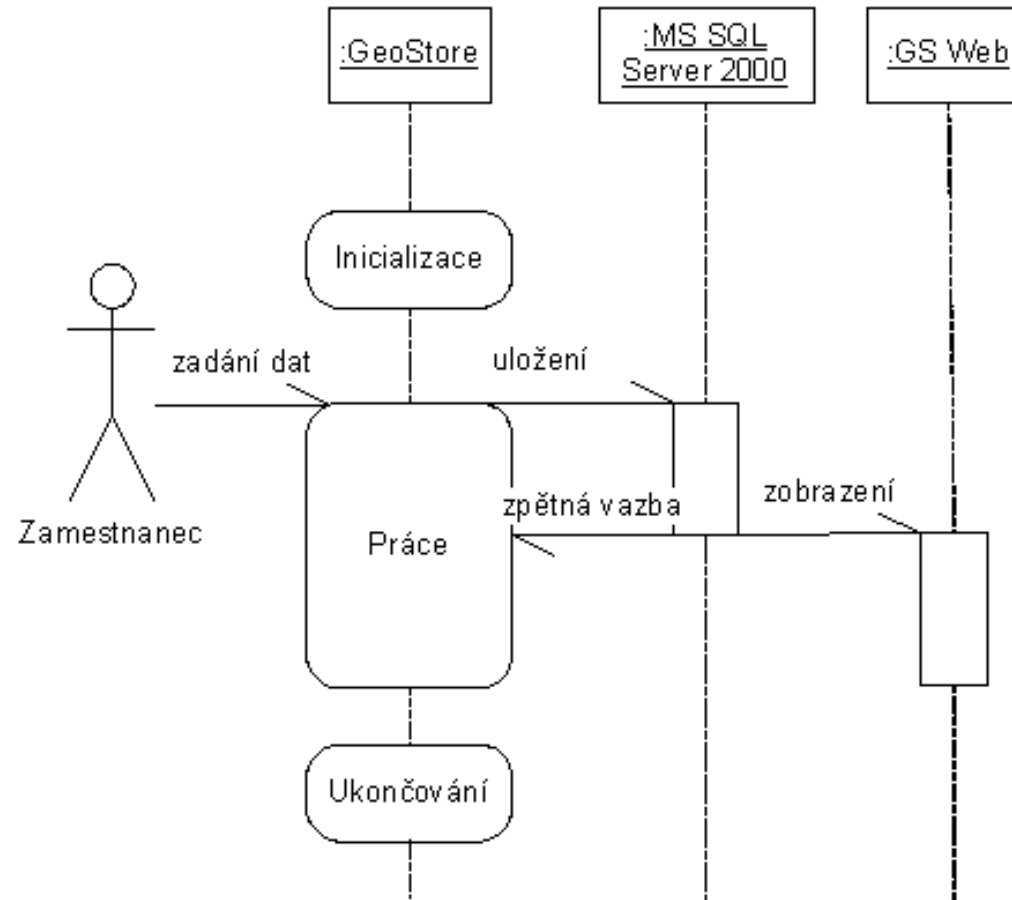


Diagram spolupráce UML

Vlastnosti diagramu spolupráce:

- Ukazuje tytéž informace jako sekvenční diagram s ohledem ne na čas, ale na propojení mezi objekty.
- Slouží pro pozdější určení vztahů mezi objekty:
 - datové vztahy - dány distribucí informací,
 - komunikační vztahy - dány spoluprací mezi objekty

Diagram spolupráce UML

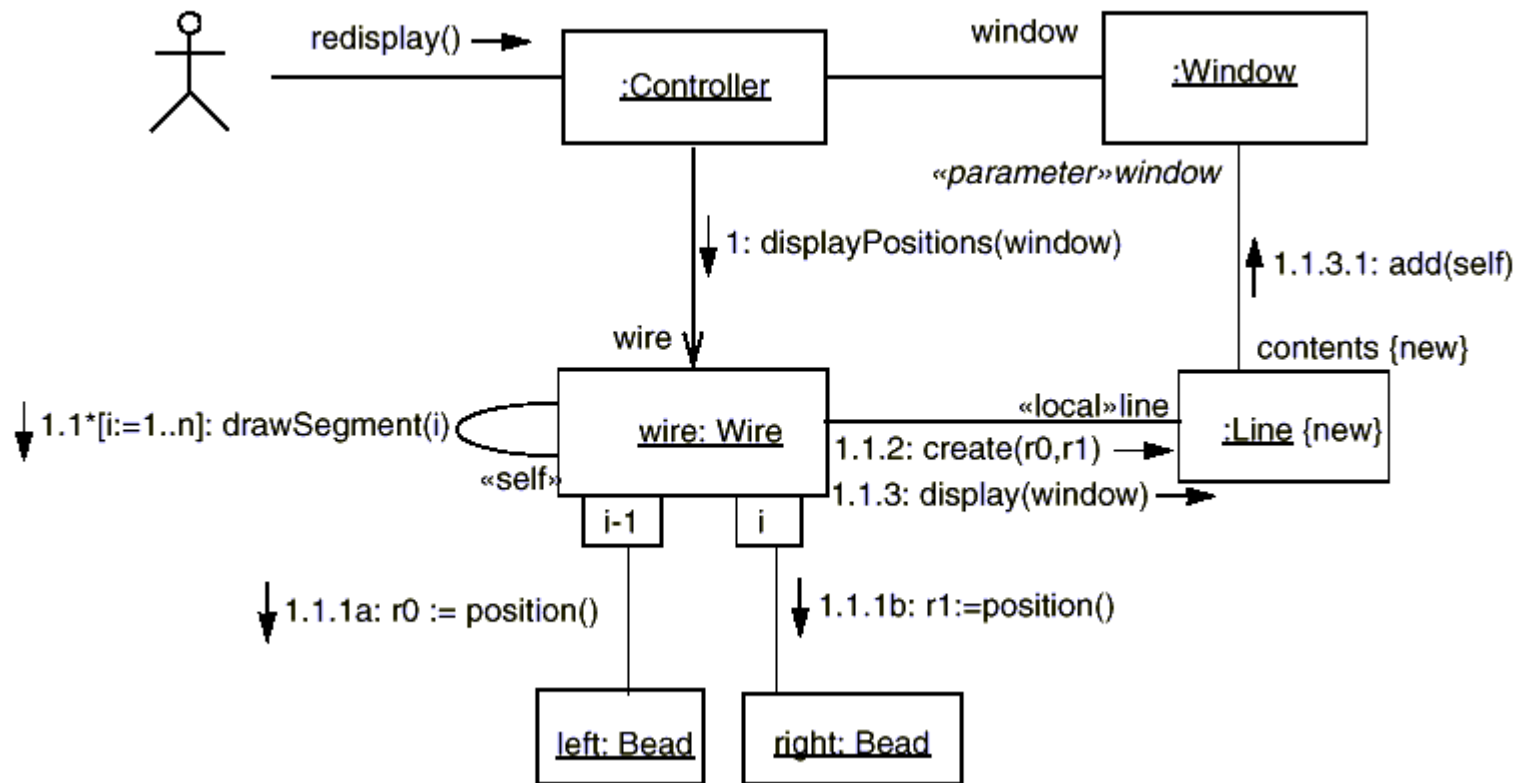


Diagram tříd UML

- Třída je abstrakce objektů, které mají společné chování a o kterých nás zajímají stejné informace.
- V OOP je to šablona pro instance objektů.
- Statický pohled na modelovaný systém.
- Vytváří se v etapě analýzy a postupně se zpřesňuje, je základem pro implementaci a nástrojem pro dokumentaci.

Diagram tříd UML - Třída

Třídy vyhledáváme analýzou problémové domény (podstatná jména ze scénářů).

Třída je zapouzdřením určitého chování a určitých informací.

Zapouzdření je koncept, který dává ve třídě dohromady to, co spolu souvisí a dává nějaký smysl.

Obsahem třídy je:

Jméno,

Atributy,

Operace.

Diagram tříd UML –Operace, atributy

Operace třídy

- Operace, které třída definuje, představují její chování nebo také zprávy, kterým třída rozumí.
- Zdrojem pro hledání operací jsou především scénáře use case analýzy.

Atributy třídy

- Atributy třídy jsou informace, které o třídě uchováváme.
- Zdrojem pro atributy třídy jsou věcné znalosti o dané problematice a analýza podrobných požadavků uživatelů.
- Atributy třídy by měly být atomické a nedělitelné.

Vztahy mezi třídami

Třídy nejsou v systému osamocené, jejich objekty ke svému chování potřebují využít schopností jiných objektů. Třídy mezi sebou sdílí informace.

Asociace (“Slabá” vazba mezi třídami), např. čtenář a kniha

- Neříká nic jiného, než to, že dvě třídy mají mezi sebou vztah, tedy že o sobě vědí.
- Defaultně obousměrná vazba.
- Může být definováno jméno asociace, role a násobnost (kolik instancí třídy existuje vůči jiné třídě).

Agregace (volná vazba mezi třídami), např. počítač a periferní zařízení

- Představuje vztah skládání celku z částí, celek odpovídá za vytvoření a zrušení částí, je to vztah celku k jedné části, definujeme násobnost, jméno a role ne.

Vztahy mezi třídami

Kompozice (nejsilnější forma asociace, velmi pevná vazba mezi třídami), např. faktura a řádek faktury, třída se skládá z jiných závislých tříd

- Třídy tvoří hierarchii

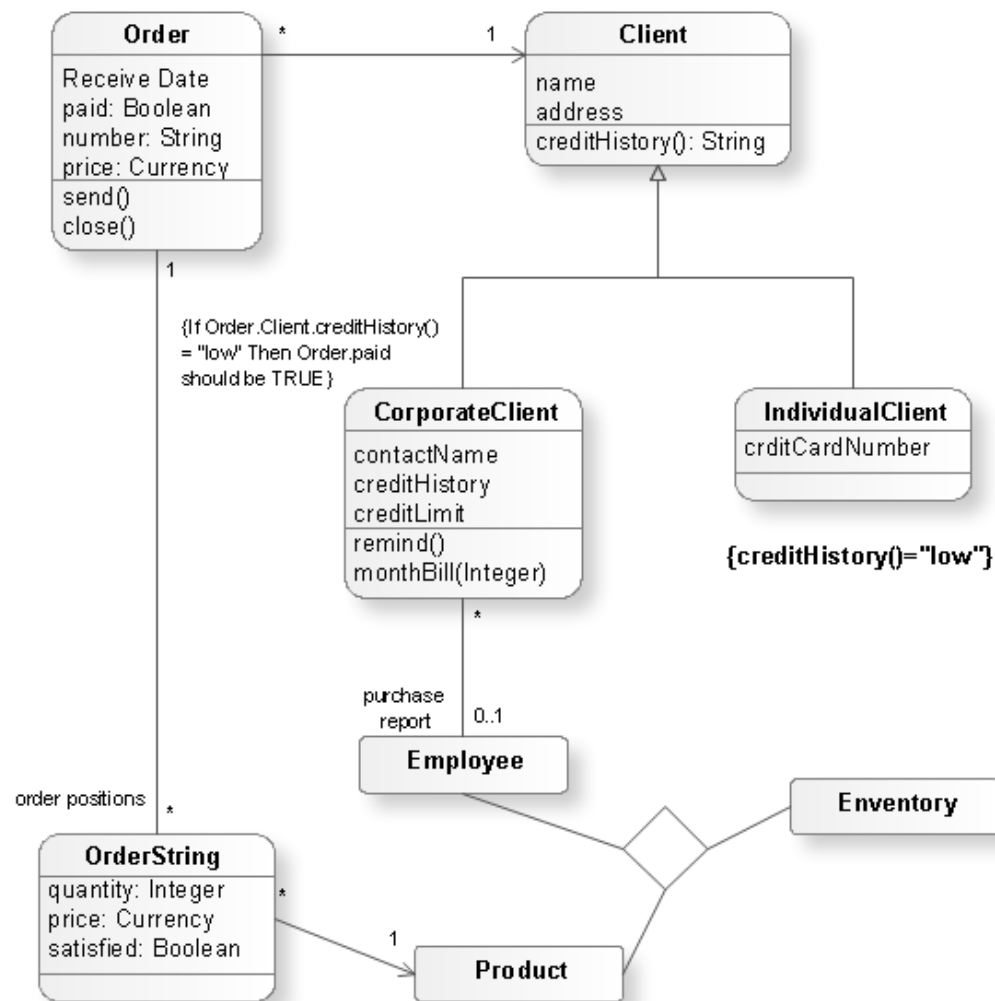
Dědičnost (nejsilnější forma vazby mezi dvěma a více třídami).

- Generalizace/specializace
- Potomek dědí celou specifikaci svých předků (atributy i operace).
- Viditelnost prvků určuje, jak jsou děděny (public a protected jsou v potomkovi přístupné, private ne).
- Vztah mezi třídou a speciálním případem této třídy.
- Rozlišuje, co je stejné a co jiné.

Vztahy mezi třídami

- **Rekurzivní asociace** (asociace na sebe sama).
- **Asociativní třída** (atributy asociace)
 - Pokud sama asociace nese určité informace, které nemohou být atributy ani jedné z asociovaných tříd.
 - Hovoříme o „link class“.
 - Může mít atributy i operace.
 - Např. vztah mezi osobou (jméno, rcislo), firmou (název) je vazba s asociativní třídou pracovní poměr (datum nástupu, funkce, plat).

Diagram tříd



Stavový diagram UML

Dynamické chování systému je modelováno pomocí diagramů aktivit i stavových diagramů

Stavový diagram

- Používá se k modelování životního cyklu jednoho objektu. Hovoříme o objektech s výrazným dynamickým chováním nověji **reaktivní objekty**.
- Stavový diagram modeluje chování systému **napříč všemi use casey**. Znáznorňuje, jak se stavy objektu mění v závislosti na událostech, které se ho dotýkají
- Stav objektu je dán hodnotami jeho atributů.
- Stav objektu může ovlivňovat jeho chování.
- Stav objektu je zachycen na stavovém diagramu jako stav jednoho objektu jedné třídy bez vazeb na jiné objekty nebo jiné třídy.

Stavový diagram

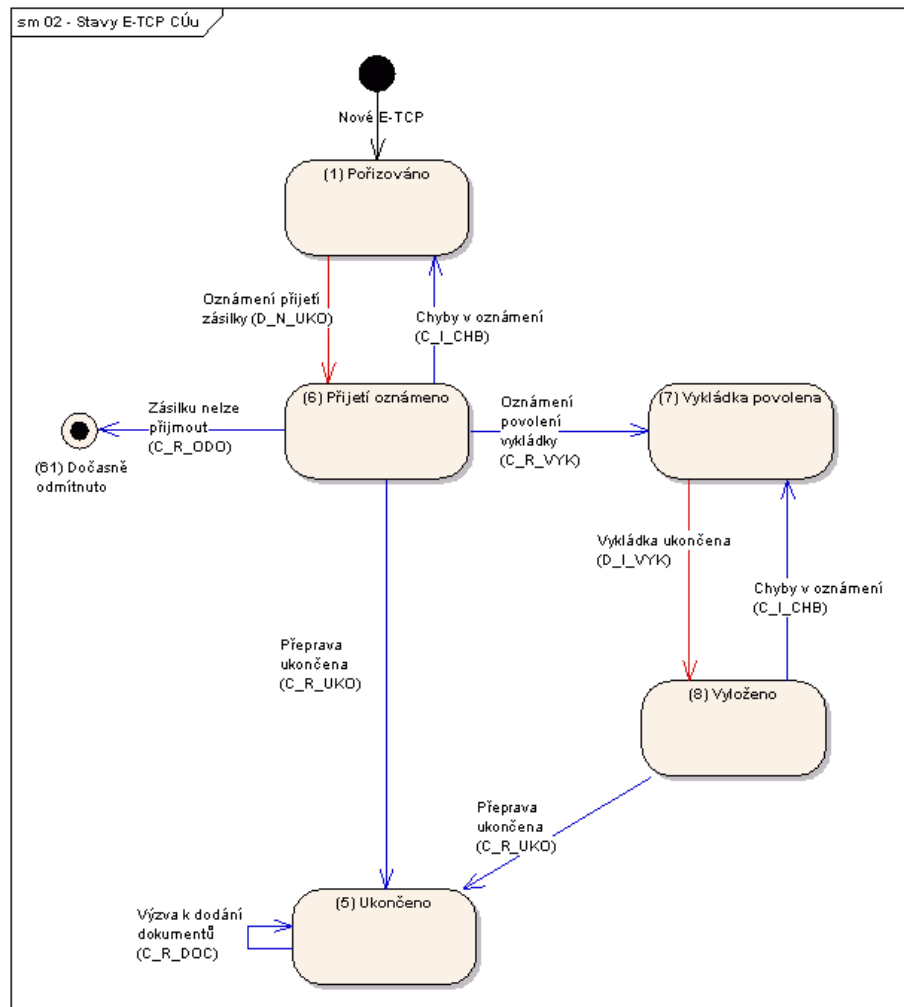


Diagram aktivit UML

- Použití pro modelování systémů pracujících v reálném čase, systémů pro řízení technologických procesů, nebo paralelních procesů a jejich synchronizaci.
- Další použití pro znázornění složitého scénáře a doplnění sekvenčního diagramu.
- Jsou zvláštním případem stavových diagramů, kde stavy jsou vyjádřeny jako akce a kde přechody jsou spouštěny automaticky po ukončení předchozích akcí nebo aktivit. Používají obvykle pouze malou podmnožinu bohaté syntaxe stavových diagramů UML.
- Lze používat symbolů rozhodování (tzv. hodnocení přechodů), symbolů rozvětvení (jeden vstup několik výstupů), spojení (více vstupů jeden výstup), plavecké dráhy – swimlanes pro specifikace osob, oddělení nebo tříd zodpovědných za aktivitu.

Diagram aktivitas

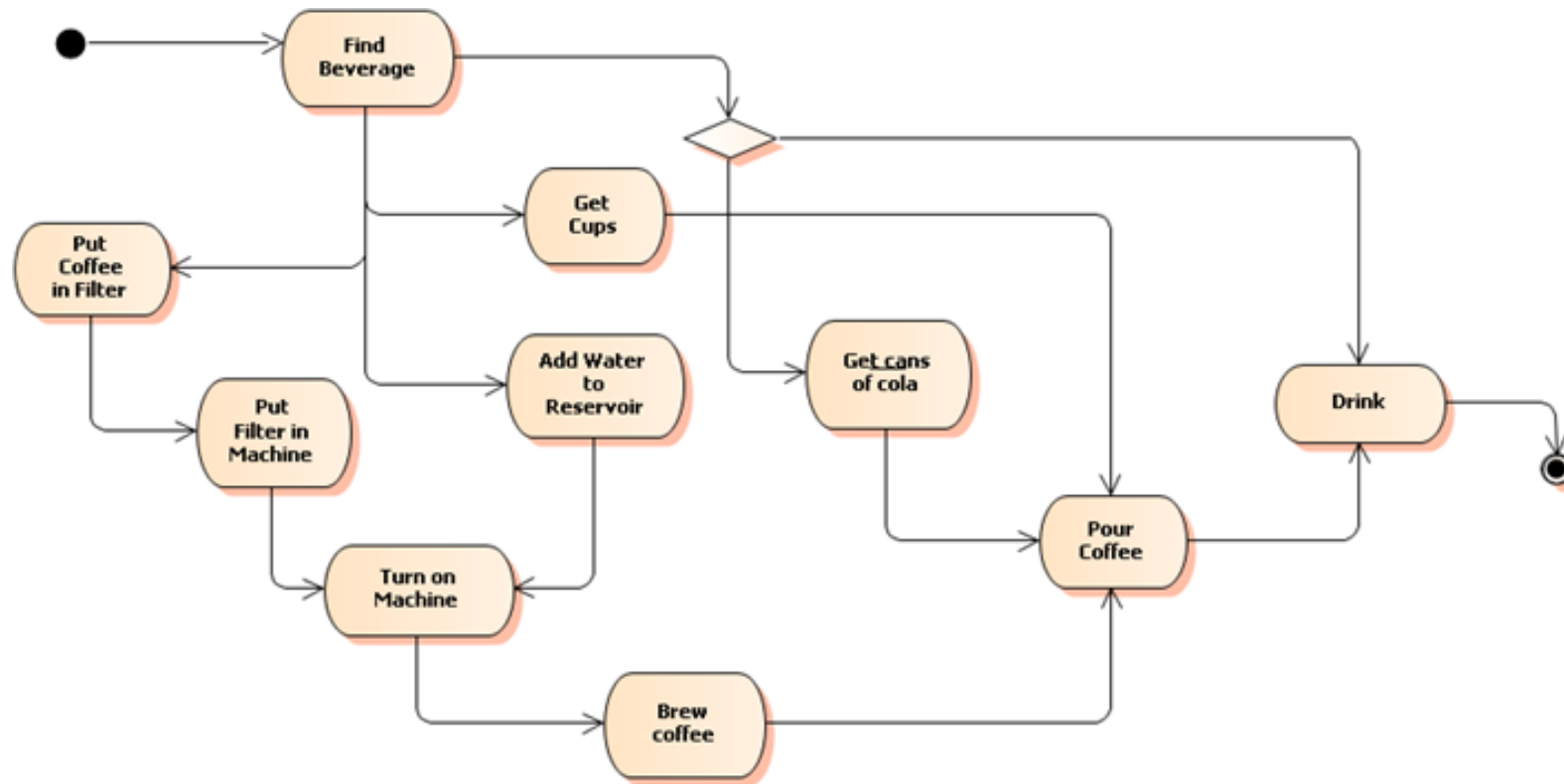


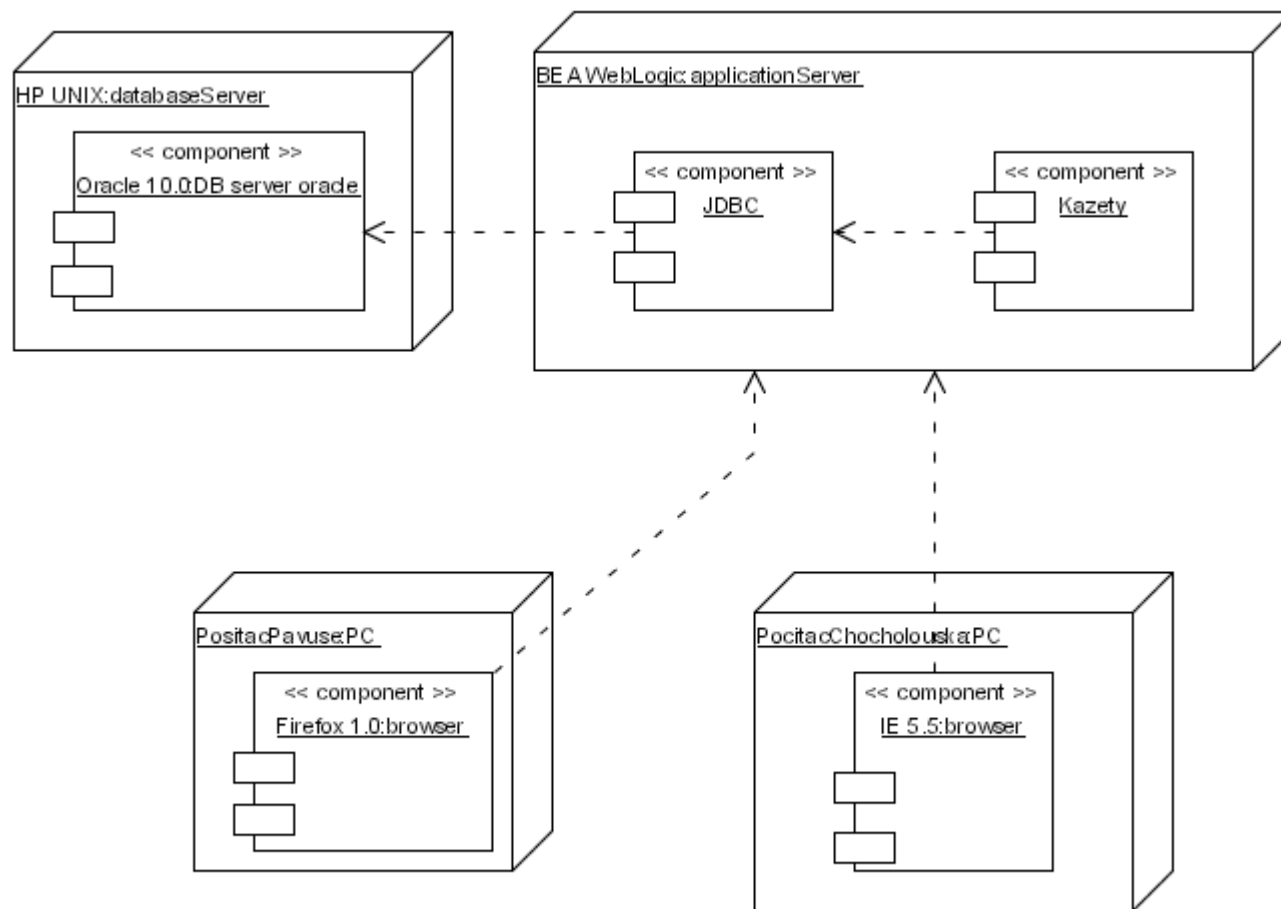
Diagram nasazení

- Etapa Nasazení (implementační etapa) je proces přiřazení artefaktů (soubory, skripty, DB tabulky, modely UML) uzlům (např. PC, server, prostředí zpracování).
- Diagram nasazení umožňuje modelovat distribuci SW systému na fyzickém HW. Ukazuje fyzický hardware na němž bude softwarový systém (komponenta) spuštěn a také způsob, jak je SW na tomto HW nasazen.

Diagram nasazení ukazuje:

- **Uzly** – typy HW, na nichž bude systém spuštěn (osobní PC, server).
- **Relace** – typy spojení mezi uzly (komunikační kanál, který slouží k přenosu informací, např. HTTP).
- **Komponenty** – typy komponent nasazených na určité uzly (modul IS, MS Word).

Diagram nasazení



Created with Poseidon for UML Community Edition. Not for Commercial Use.

Podnikové IS

Prvky IS

subsystém 1 – lidé

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

subsystém 2 – 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

Prvky IS

subsystém 3 - 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

Typy IS

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

IS organizací

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

1. podpora řídicích a administrativních funkcí

- **systemy na podporu provozu (chodu) firmy** - provozní, transakční IS - ERP - enterprise resources planning
- **systemy na podporu rozhodování** - MIS - management IS, EIS - executive IS, BI - business intelligence
- **systemy na podporu plánování** - APS - advanced planning and scheduling, SCM - supply chain management, HR - human resources
- **systemy 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-byznys
- 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.

TPS a jeho složky

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

Enterprise Resource Planning (ERP systemy)

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.

Historie ERP (Enterprise Resource Planning)

- 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.

Historie ERP (Enterprise Resource Planning)

- 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“.

Historie ERP (Enterprise Resource Planning)

- 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 kriteriálních 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.

Požadavky na ERP v EU

- 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
- Přejít na euroměnu, obchodování v Euro
- Zánik lokální měny, konverze na Euro

ERP v ČR

ERP lze rozdělit dle funkcionality na 2 základní typy:

- All-in-One
- 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 funkcionality, nákladná customizace
Best-of-Breed	Orientace na specifické procesy nebo obory, nepokrývá všechny klíčové procesy	Špičková detailní funkcionality, nebo specifická oborová řešení	Obtížnější koordinace procesů, nekonzistentnost v získávání informací, nutnost řešení více projektů

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.

Best-of-Breed ERP

- IS VEMA – zaměření na ekonomiku a personalistiku
- IS FEIS – oblast ekonomiky, logistiky a obchodu (středně velké firmy)

TRENDY VE VÝVOJI ERP SYSTÉMŮ

- Dřívějším trendem v oblasti ERP bylo dosažení maximální funkcionality.
- S rozšiřováním funkcionality ale rostla složitost systémů.
- Nynější trend, označovaný jako ERP druhé generace, je **integrace**. ERP si ponechává pouze funkce, pro které byl primárně určen – tj. podpora podnikových procesů. Další funkce se řeší integrací se specializovanými produkty (reportovací nástroje, oblast workflow, atd).

TRENDY VE VÝVOJI ERP SYSTÉMŮ

PODPORA UŽIVATELŮ

- Do této oblasti patří například nástroje business intelligence. Základním úkolem BI je monitorovat, analyzovat a plánovat podnikové procesy.
- Analytické nástroje na rozdíl od reportingu nezobrazují pouze stav hodnot, ale 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

TRENDY VE VÝVOJI ERP SYSTÉMŮ

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.

TRENDY VE VÝVOJI ERP SYSTÉMŮ

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.

TRENDY VE VÝVOJI ERP SYSTÉMŮ

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 vlastní 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

MODELY DODÁNÍ ERP

- **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

- 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

- **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

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.

VÝZNAMNÍ VÝROBCI ERP

Infor

- Získal formou akvizicí 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.

VÝZNAMNÍ VÝROBCI ERP

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

VÝZNAMNÍ VÝROBCI ERP

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

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).

SAP

- 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 zaměřeny 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.

Úvod do teorie systémů

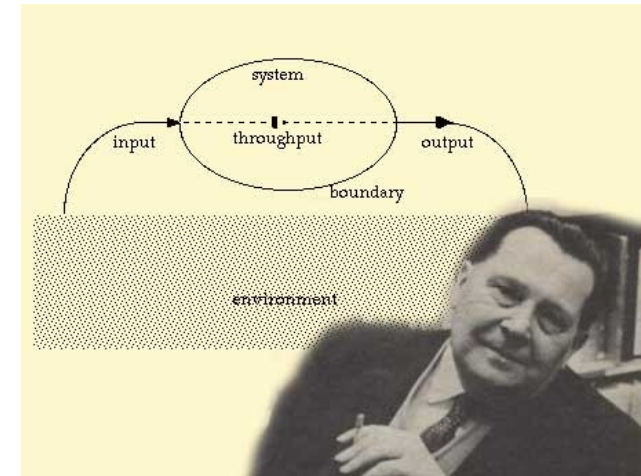
(od přírodních věd k vědě informační)

Mgr. Marek Timko, Ph.D.

Systemový přístup

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

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



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

System a struktura

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

„Wikidefinice“: „*System (česky soustava) je souhrn souvisejících prvků, sdružený do nějakého smysluplného celku. System 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.

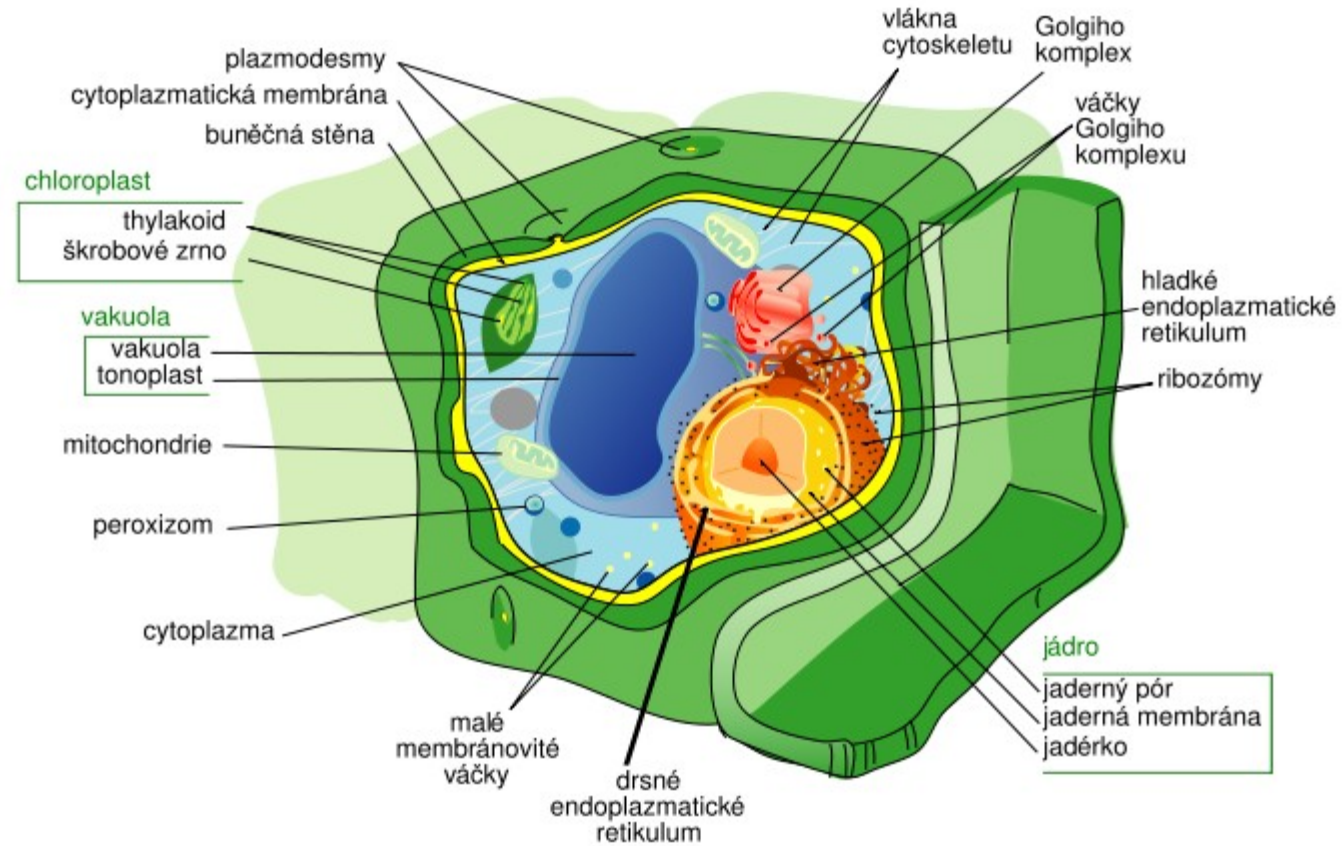
A co *emergence*? Odkud se vynořuje? Kolik neuronů myslí?

System

System vs. objekt/jsoucno

- na objektu lze pozorovat množství (sub)systemů (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.

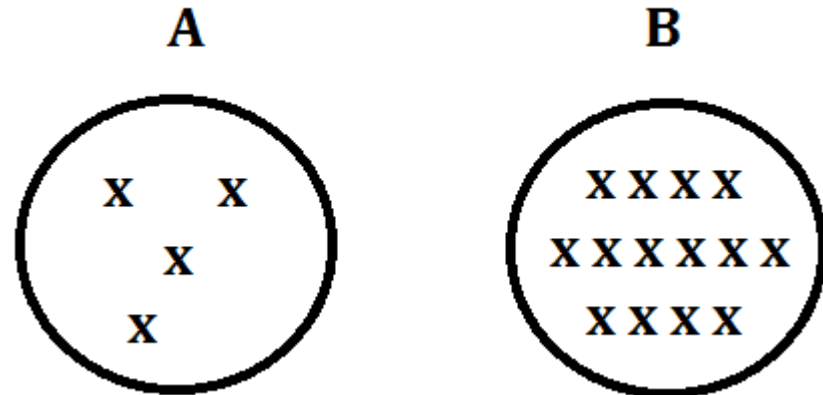
(Cybernetics or the Control and Communication in the Animal and the Machine, 1948)

Jak komunikují živé systémy a jak systémy technické?

Složitosti systémů

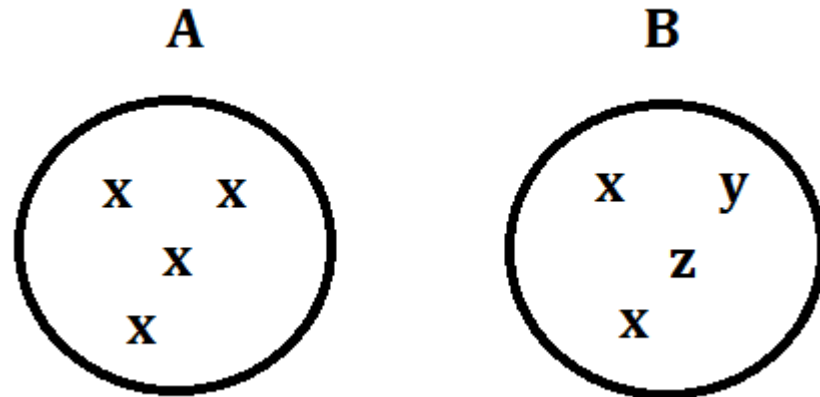
Diferenciace komplexity:

1.) počet prvků



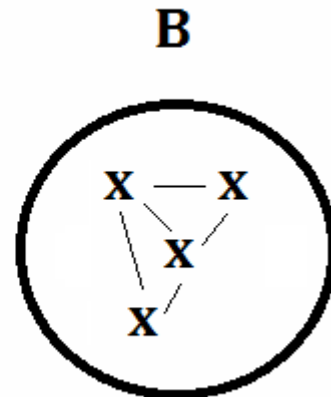
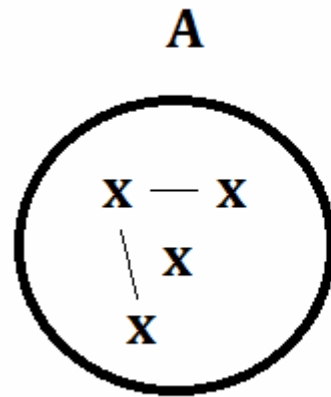
Složitosti systémů

2.) různorodost prvků



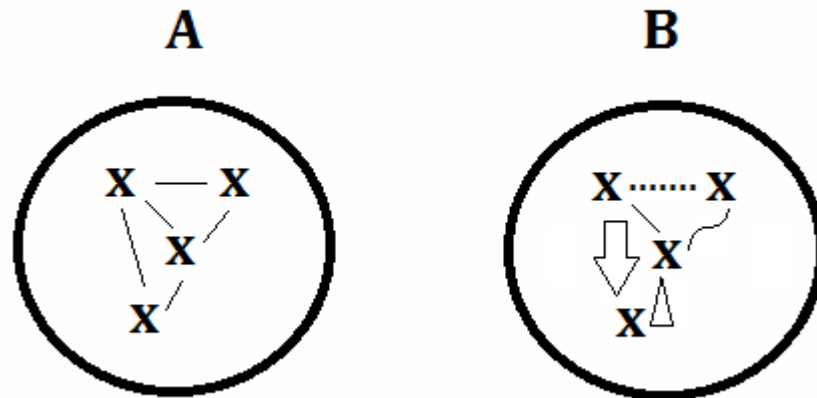
Složitosti systémů

3.) počet vazeb



Složitosti systémů

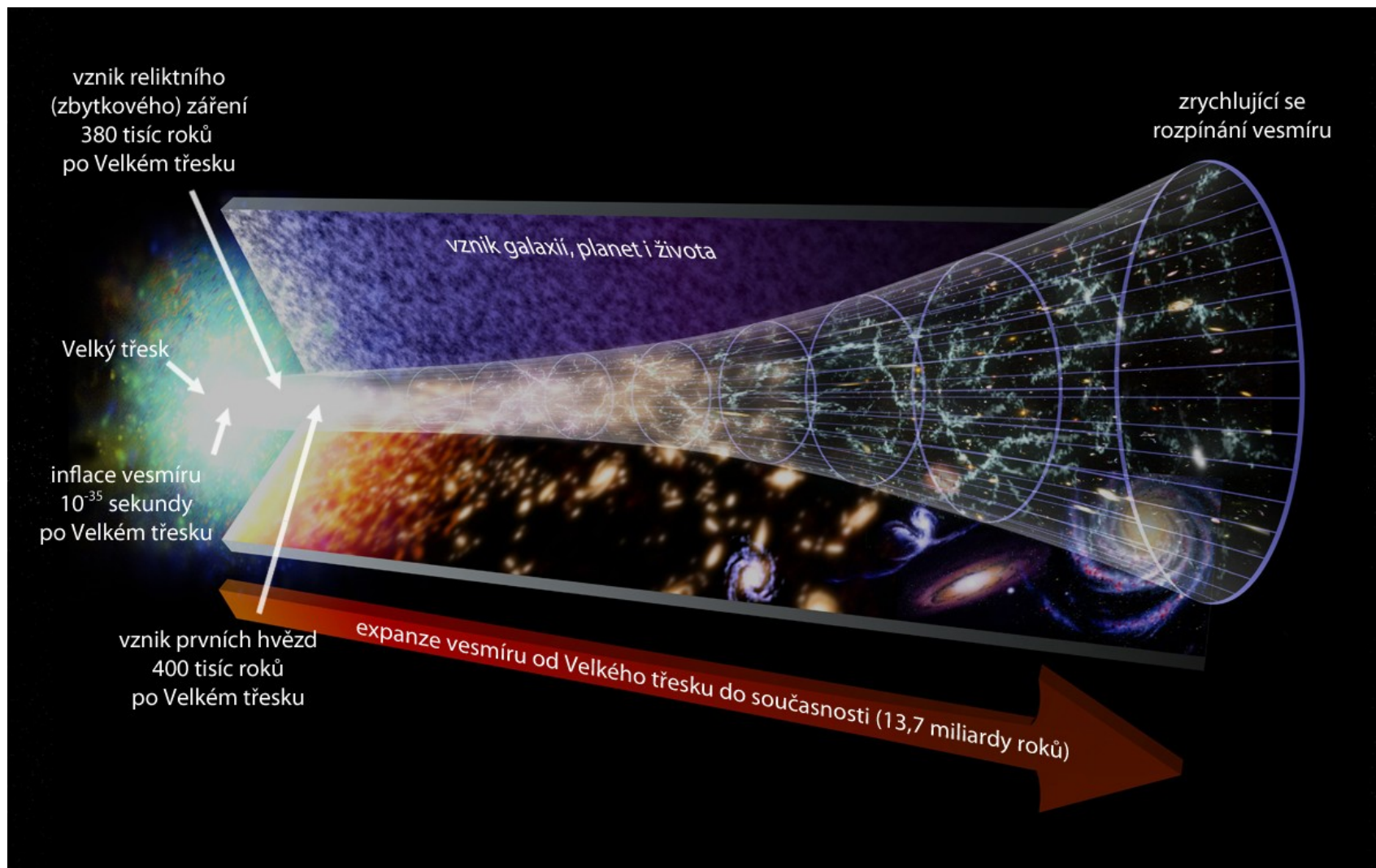
4.) různorodost vazeb



Typy systémů:

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

Izolovaný systém?

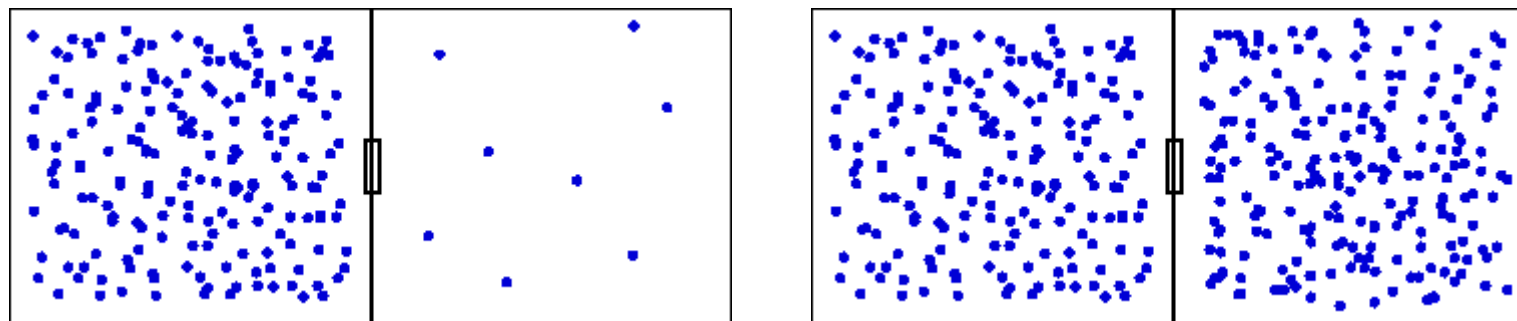


Systemy a evoluce („prokletí“ entropie!)

A jak to všechno souvisí s informací?

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

Uff, ta prokletá *entropie*!



Negentropie pana Schrödenigera

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.“*



Erwin Schrödinger (1887–1961)

Systemy a chaos (nelineární disipativní struktury)!

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

„Neexistuje jednoznačný průběh vývoje.“

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

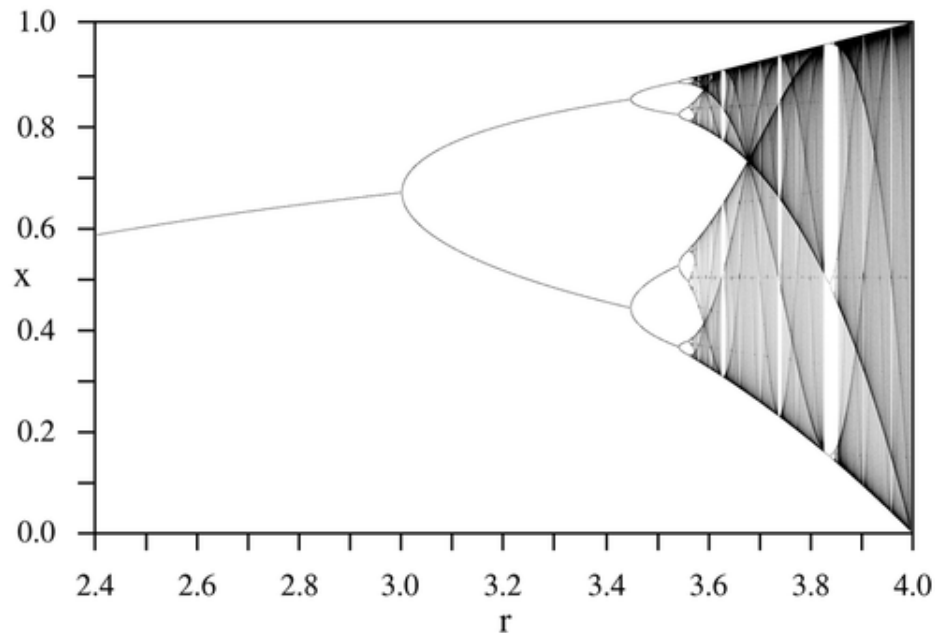
„Růst nevratně probíhající v čase vytváří nevratný vývoj.“



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!)



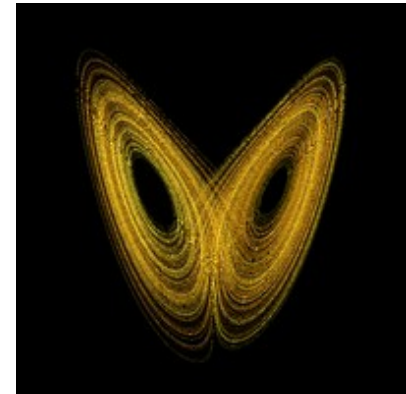
Deterministický chaos

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 počátečné 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 Santiago: 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!

