

2

Unified Modeling Language

Marek Tavač
Ján Ružbarský

Zdroje a literatúra

- www.omg.org
- UML distilled third edition –
Martin Fowler with Kendall Scott
- Learning UML – Sinan Si Alhir
- Object Oriented analysis and Design Using UML –
Mark Collins
- Myslíme v jazyku UML – Joseph Shmuller
- Design Patterns – Erich Gamma a kol.,
-

Unified Modeling Language

- Módny trend
- Štandard
- Lingua franca

Čo je UML ?

- jazyk určený pre špecifikovanie, vizualizáciu, konštrukciu a dokumentovanie súčastí **softvérových systémov** ako aj pre **business modeling** alebo **modelovanie iných nesoftwareových systémov**
- je vizuálny jazyk na modelovanie a komunikáciu o systéme pomocou diagramov a podporného textu
- UML 2.4.x

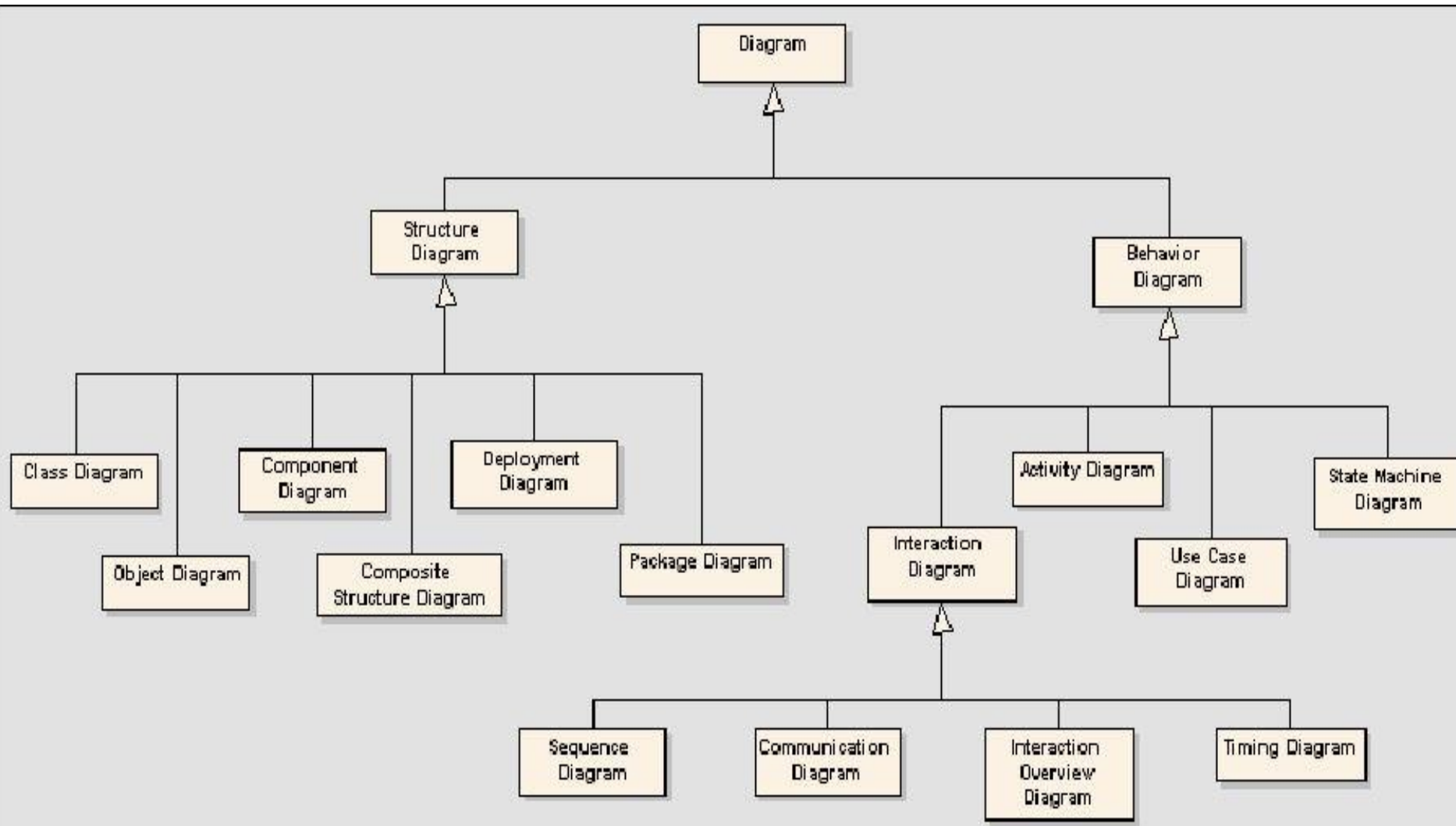
- Pomáha špecifikovať, vizualizovať, dokumentovať modely softvérových systémov
- Spôsob zápisu, nie metóda, metodológia na aplikovanie notácie
- Diagramy (modely)
 - každý projekt iba tie, ktoré potrebuje
 - závisí to od úrovne abstrakcie popisovaného systému

- Cieľ - uspokojenie požiadaviek
- Všetky fázy vývoja systému
- Softvérové aj nesoftvérové systémy
- Nestanovuje žiadny osobitný prístup
- Natívne podporuje hlavne OOP

Z čoho sa UML skladá ?

- **Diagramy**
 - **štrukturálne** diagramy - *statické štruktúry aplikácie*
 - diagramy **správania** - *všeobecné typy správania*
 - diagramy **interakcie** - *d'alšie aspekty vzájomného pôsobenia*
- **Poznámky**
- **Stereotypy**

Štruktúra UML diagramov



- **Systém (informačný)** – kombinácia softvéru a hardvéru, umožňujúca riešenie konkrétneho problému
- **Vývoj systému** – vytváranie systému pre klienta, osobu, ktorá má nejaký problém
- **Reuse** - znovupoužiteľnosť

Spravovanie modelov

Projekt = niekoľko pohľadov na systém

- pohľad na štruktúru
- pohľad na funkčnosť
- pohľad na nasadenie
- ...

Možnosti spravovania modelov:

- **Balíčky**
- **Subsystemy**
- **Modely**

Diagram tried - class diagram

Symbol triedy UML : štvoruholník

– **môže** mať 1 až 3 časti

- Meno triedy
- Atribúty
- Operácie



Atribúty

- Vlastnosť triedy reprezentujúca dáta
- Trieda nemusí mať atribúty
- UML : druhá časť v symbole tried – zoznam atribútov
- Atribúty triedy

Pracovník
<ul style="list-style-type: none">-DalsilD:integer=1-ID:integer-Meno:String-EmailAdresa[1..5 unordered]:String="Žiadna adresa"-Telefon[1..* ordered]:String#Heslo:String

- EmailAdresa [1..5unordered]: String = „ Žiadna adresa“

Metódy (operácie)

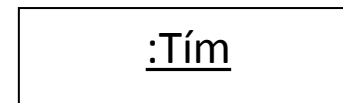
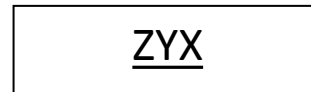
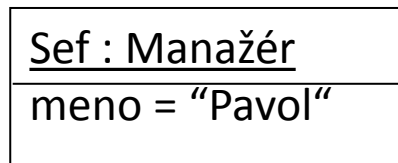
- Čo môže trieda vykonávať
- Čo s ňou môže robiť iná trieda
- Príklad: funkcie, procedúry
- Metódy triedy

Pracovník
<u>+create():Pracovník</u> +getTelefon(in Priorita:Integer):String +setTelefon(in Priorita:Integer, in Cislo:String) +pridajEmailAdresu(in Adresa:String=""):Boolean +odstranEmailAdresu(in Adresa:String) getEmailAdresa(out PrvaAdresa:String, out DruhaAdresa:String, out TretiaAdresa:String, out StvrtaAdresa:String, out PiataAdresa : String) #setHeslo(in Heslo:String) +destroy()

+ **pridatEmailAdresu** (in nováEmailAdresa:String="") : Boolean

Diagram objektov - object diagram

- Podobný diagramu tried
- Rozdiel: symbol v UML – max. 2 časti
 - Meno
 - Atribúty

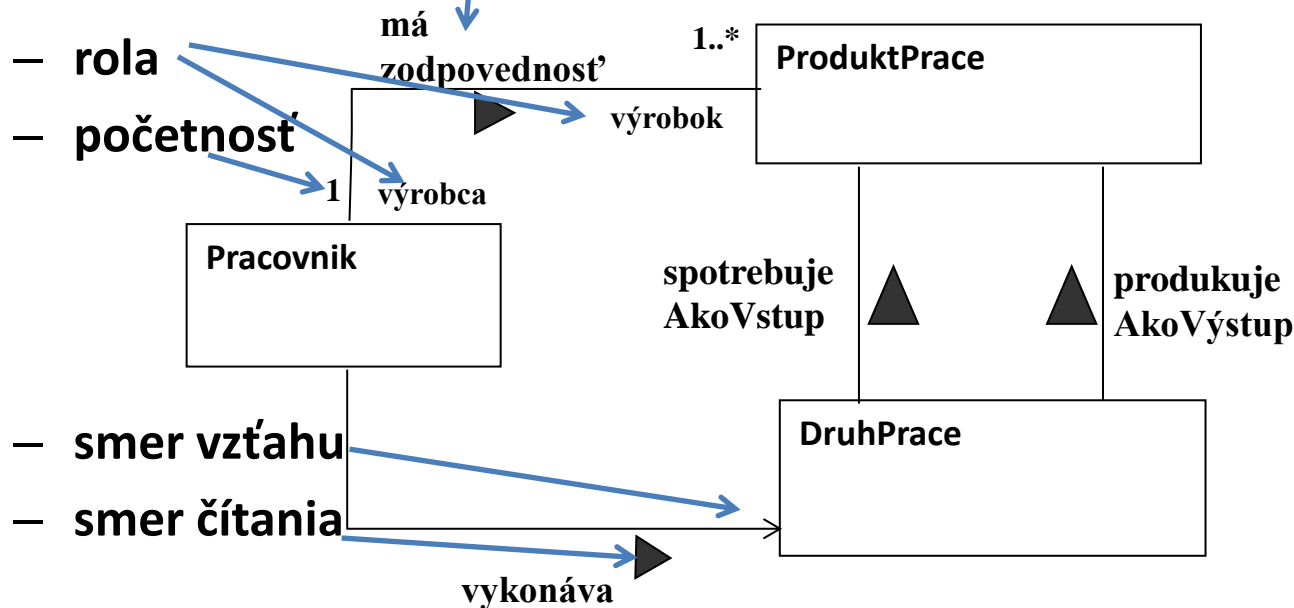


Asociácie a väzby

- vzťahy
- **Asociácia** – všeobecný vzťah medzi triedami
(v diagrame tried)
- **Väzba** – konkrétny vzťah medzi objektmi
(v diagrame objektov)

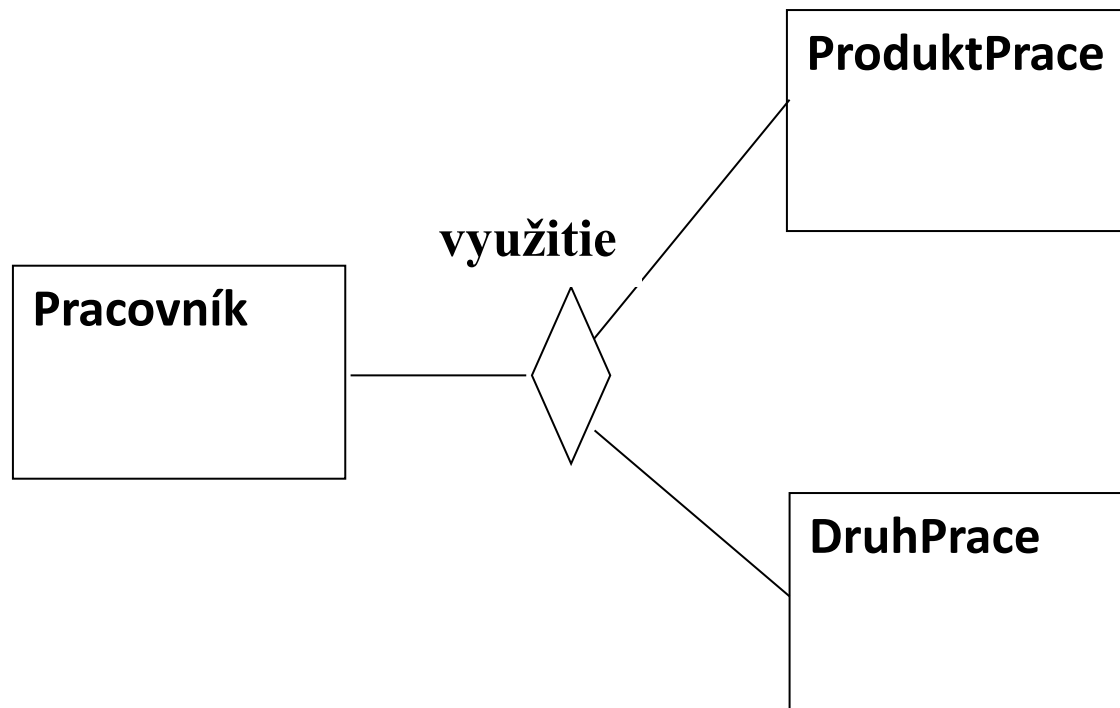
Binárna asociácia

- asociácia medzi dvoma triedami
- môže byť označená menom
- popisuje sa slovesom
- na konci



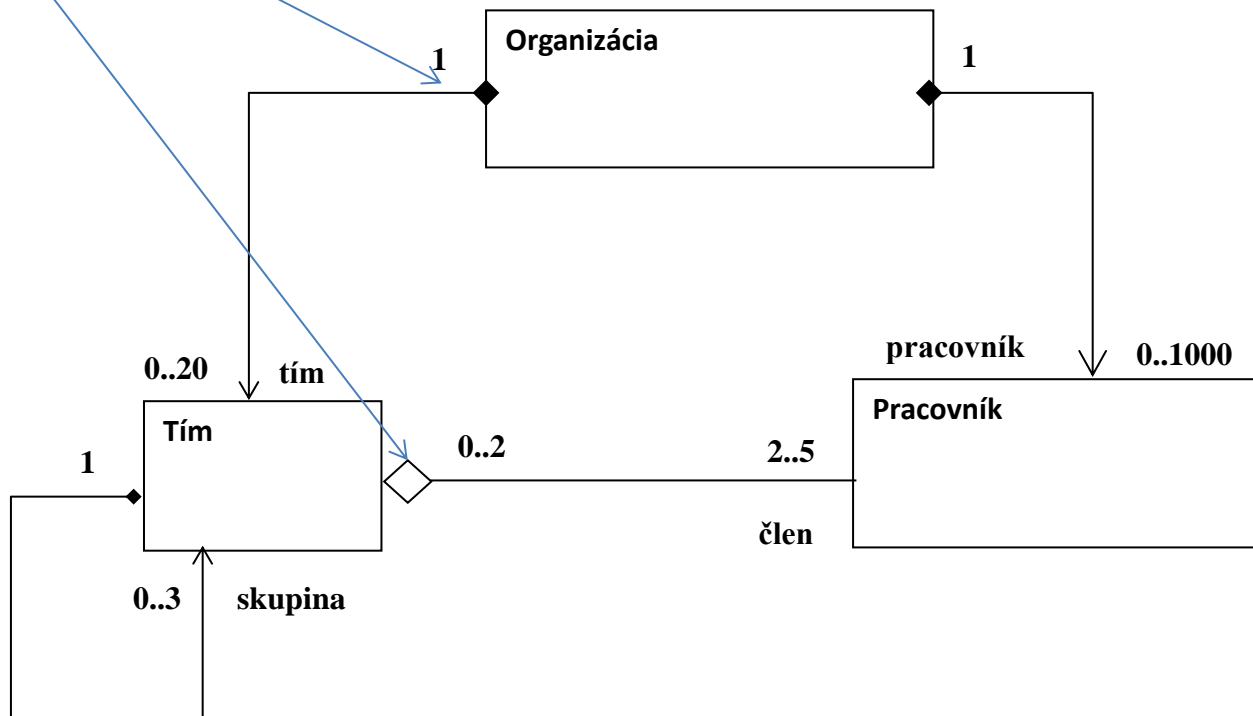
N – árna asociácia

- Vzťah medzi viacerými triedami
- Označenie slovesom alebo podstatným menom, vyjadrujúcim činnosť



Agregácia, Kompozícia

- **Agregácia** – vzťah has-a (má)
- **Kompozícia** – vzťah contains-a (obsahuje)
aj *kompozitná agregácia*



Generalizácia

- Vzťah dedičnosti
- Rodič (nadtrieda)
- Dieťa (subtrieda)
- Predok
- Potomok

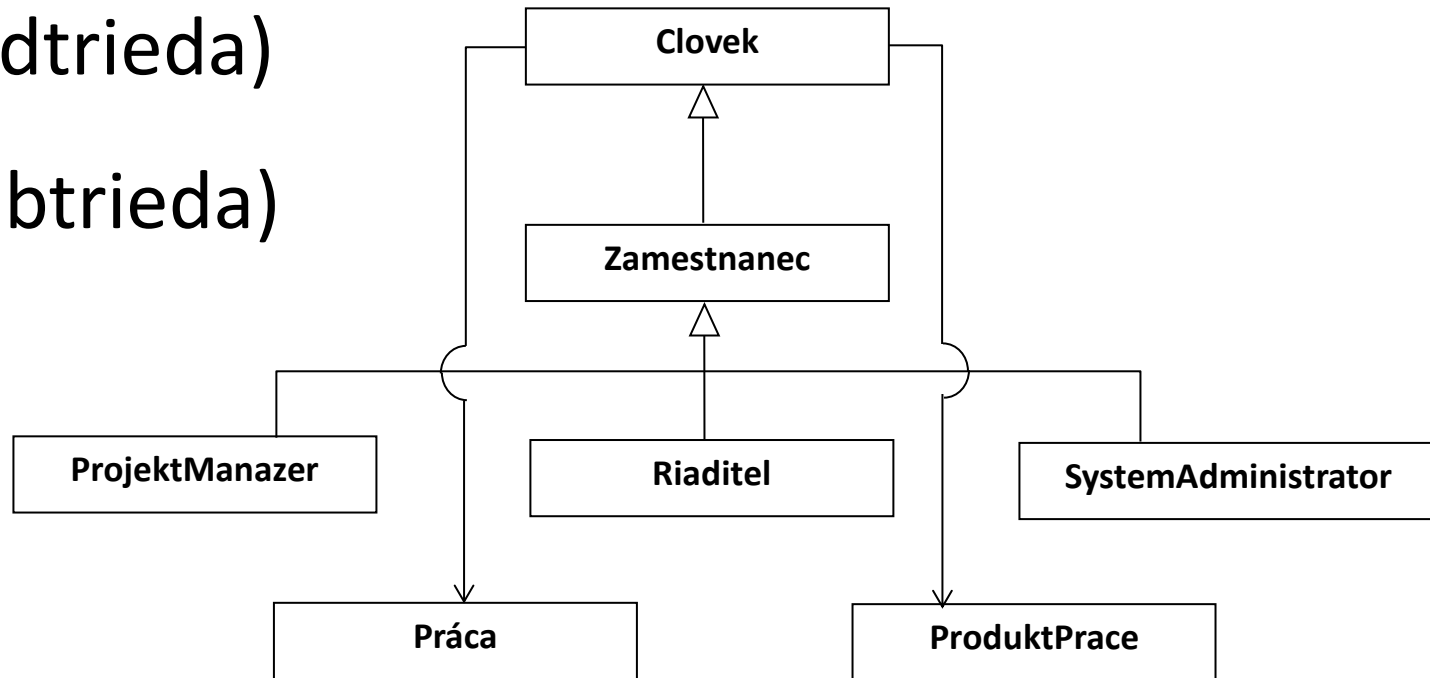
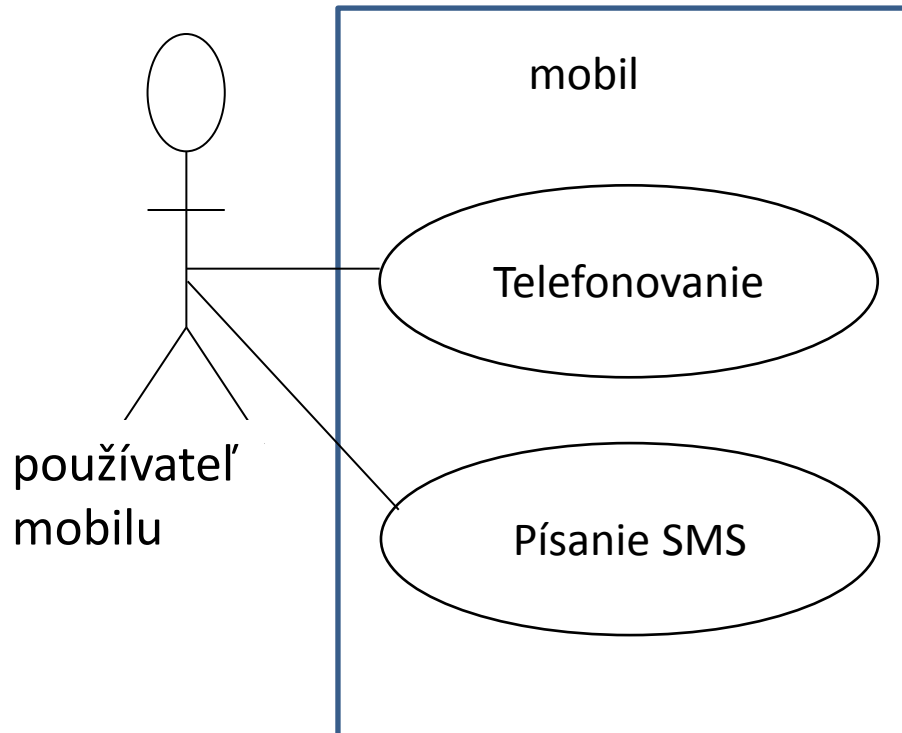


Diagram prípadov použitia – Use Case Diag.

- funkčnosť systému
- použitie
- **actor** (participant)
- **use case** – prípad použitia
- vzťahy medzi nimi
- UML – žiadny návod ako zachytiť požiadavky

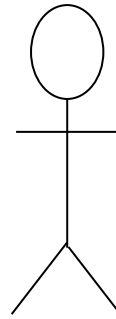
Diagram prípadov použitia - Use case diag.

- popis správania z pohľadu používateľa



Aktor, aktér

- používateľ
- externý systém
- ikona figúry alebo štvoruholník so stereotypom

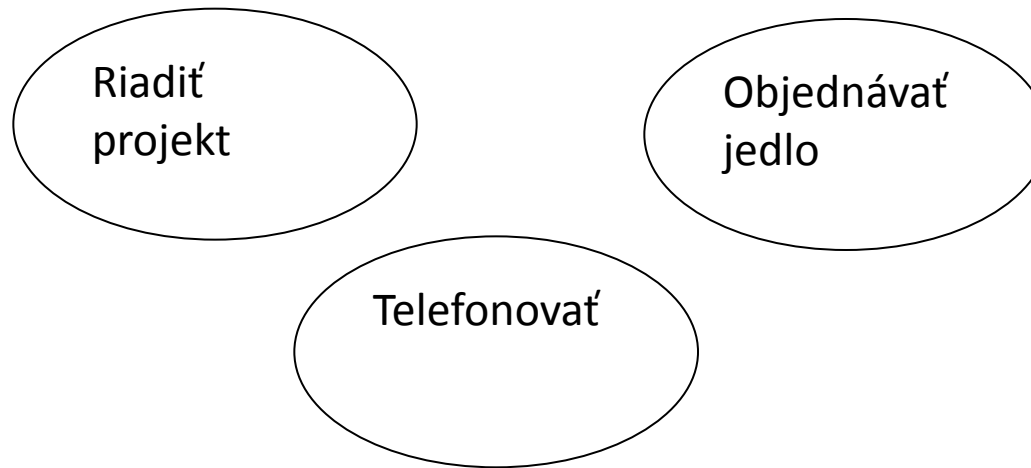


Systemový administrátor

<<actor>>
Zálohovací systém

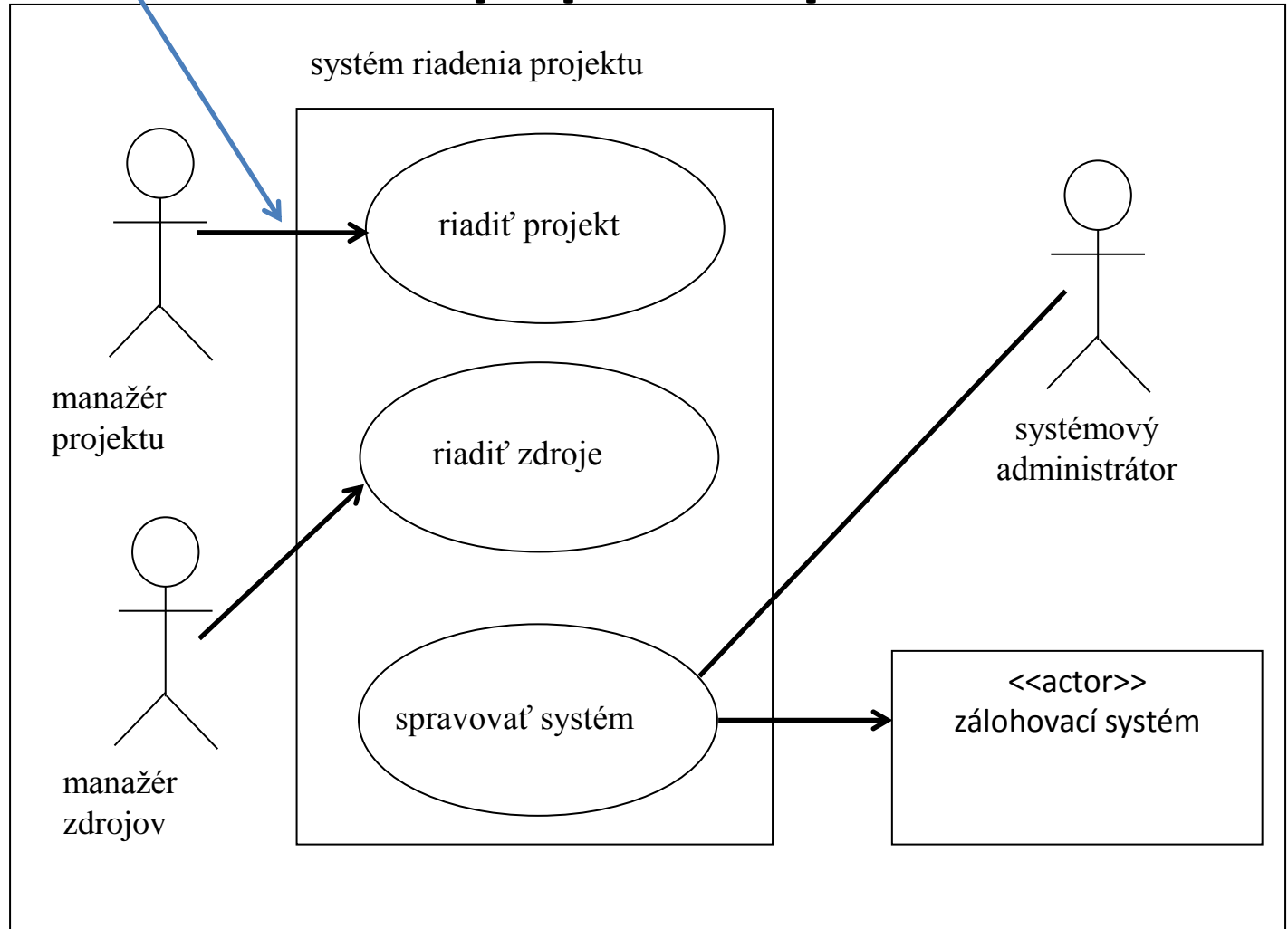
Use Case – prípad použitia

- funkčná požiadavka na systém
- požiadavky zapísané ako postupnosť krokov



Komunikačná asociácia

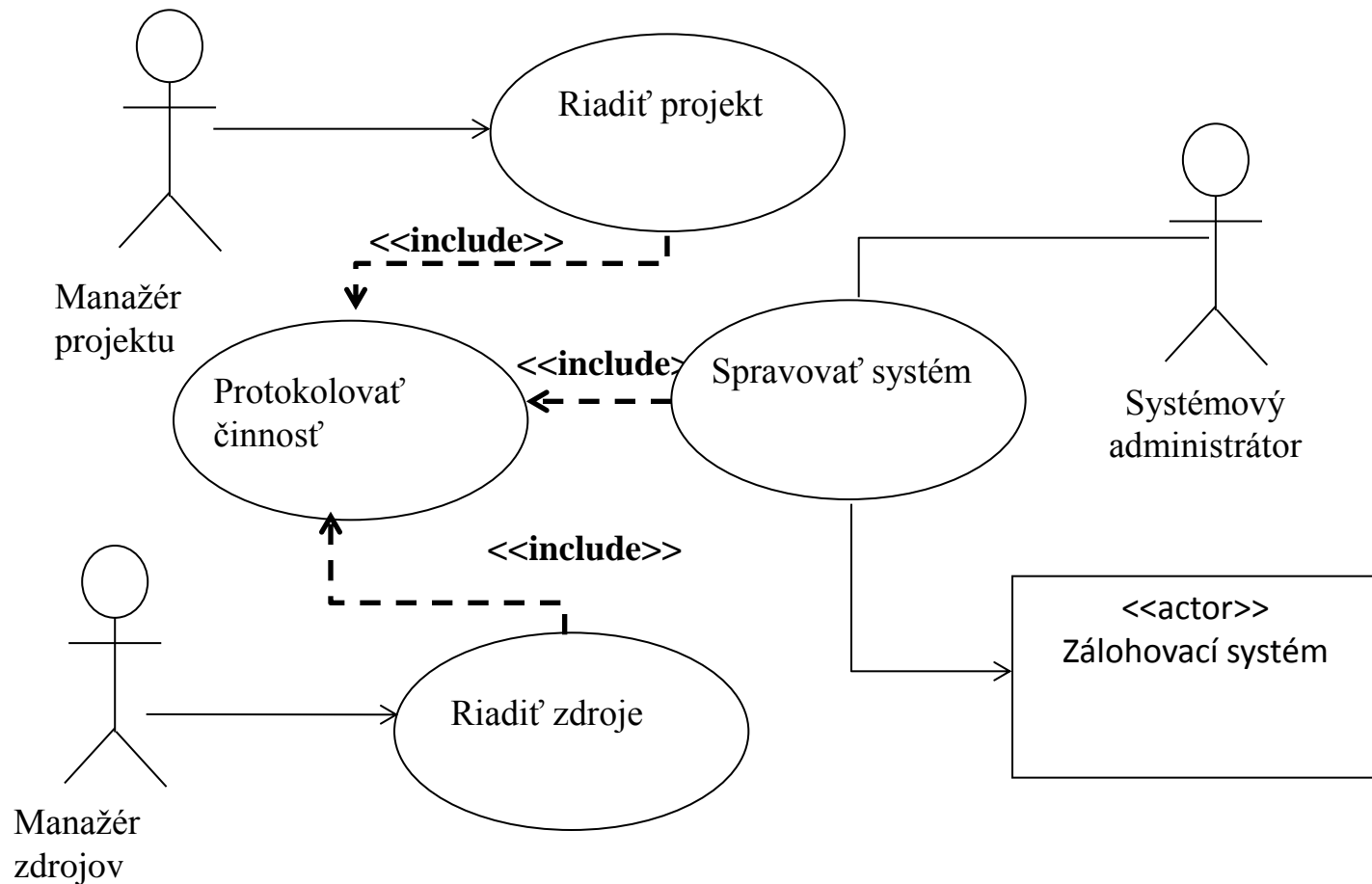
- vzťah medzi **aktorom** a **prípadosm použitia**



Include závislosť – vzťah vkladania

- vzťah medzi **prípismi použitia**
- Základný (base) use case
- Vkladaný (inclusion) use case
- Prerušovaná čiara – ***šípka od základného k vkladnému***

Include závislost - príklad



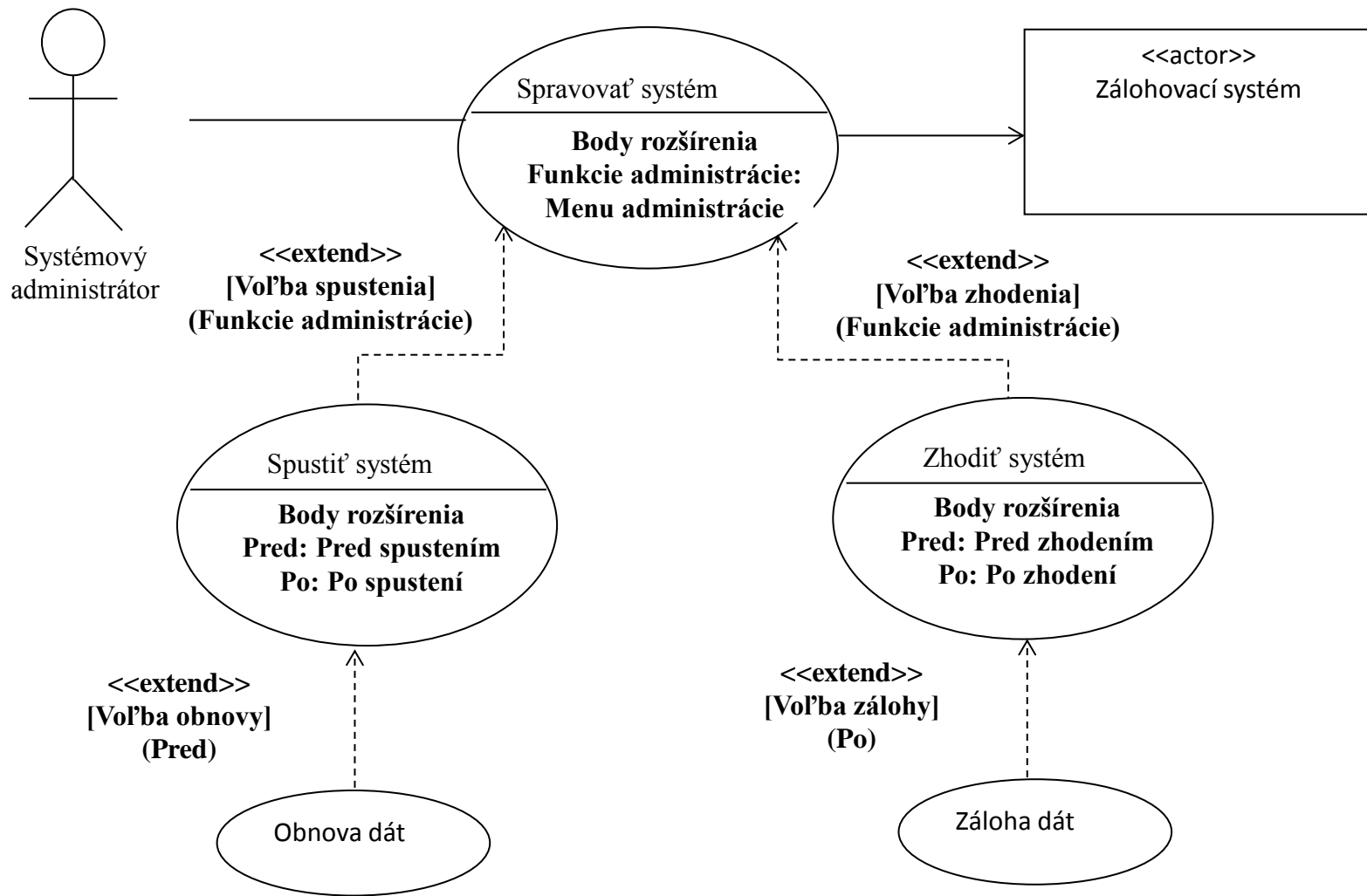
Extend závislosť – vzťah rozširovania

- vzťah medzi **prípismi použitia**
- reuse pre use case
- Vzniká doplnením ďalších krokov do základného UC
- Základný (base) use case – Riadiť projekt
- Rozšírený (extension) use case – Udržovať úlohy
- Prerušovaná čiara – ***šípka rozšíreného k základnému***

Extension points - Body rozšírenia

- Pre extend závislosť
- Rozšírenie je možné urobiť len v týchto bodoch
- Aj niekoľko v jednom use case
- Nová časť v ikone pre use case
- Popis

Body rozšírenia



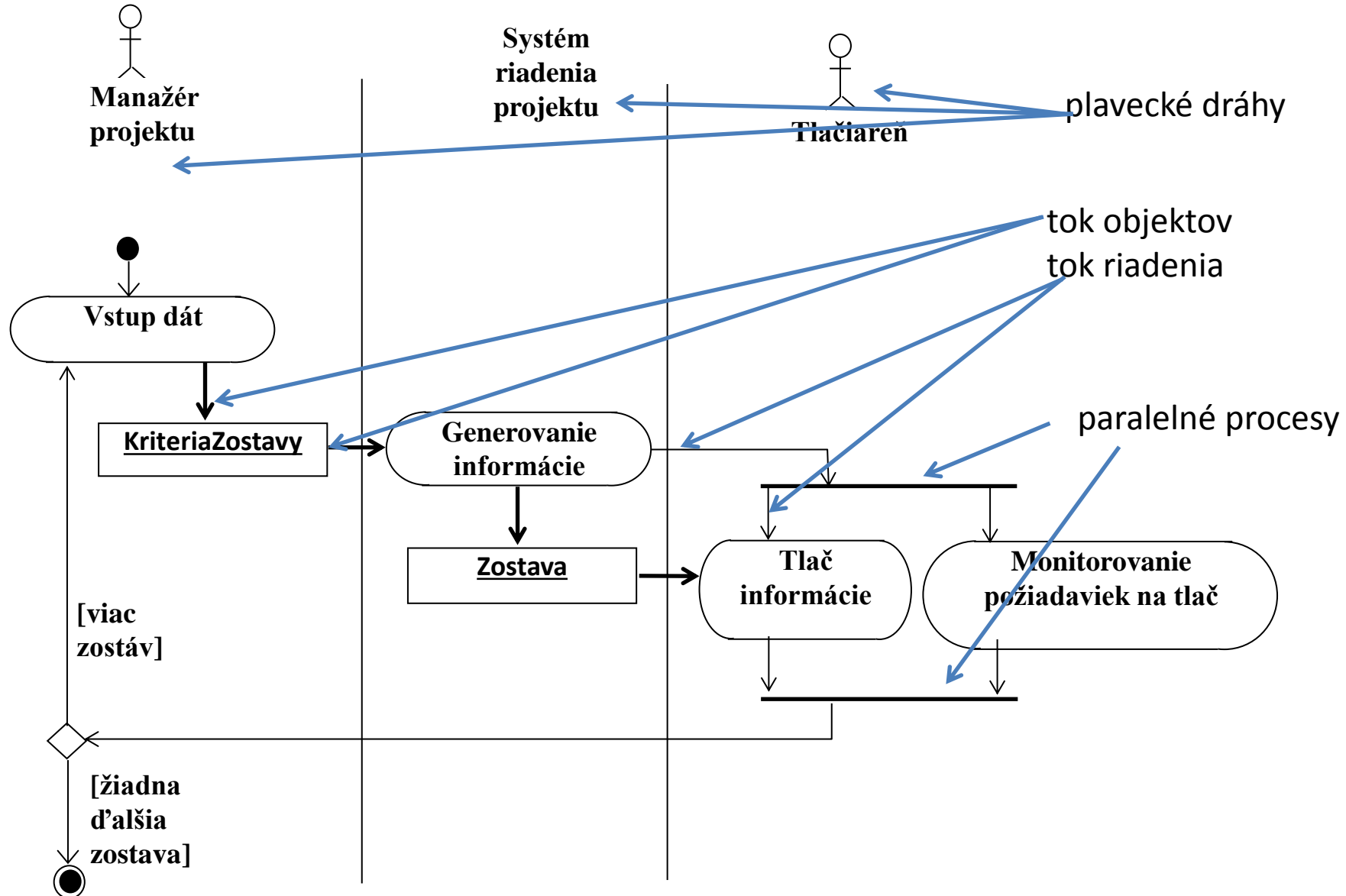
Generalizácia v UC diagrame

- medzi aktormi - **Aktorové** generalizácie – podobnosť aktorov
- medzi prípadmi použitia - generalizácie **prípadov použitia** – podobnosť prípadov použitia

Diagram činností - Activity diagram

- Další spôsob zobrazenia správania systému
- Podobný ako stavový diagram
- zobrazuje jednotlivé kroky operácie, procesu (podobný vývojovému diagramu)
- **tok riadenia**
- **tok objektov**
- usporiadanie, nasledovanie jednotlivých aktivít za sebou

Diagram činností - príklad

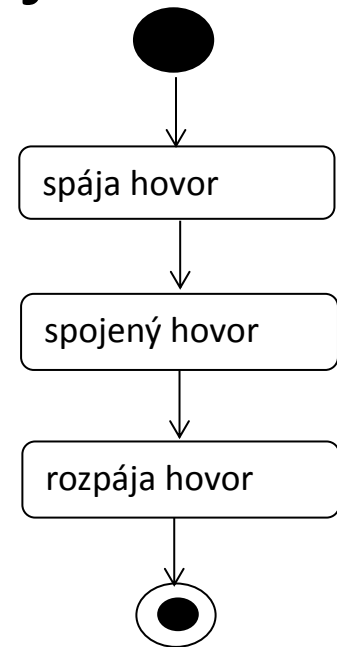


Stavový diagram - State diagram

- popisuje životný cyklus prvkov
- správanie prvku systému v čase
- časové zmeny UML modelu

Stav = prirodzená vlastnosť všetkých objektov

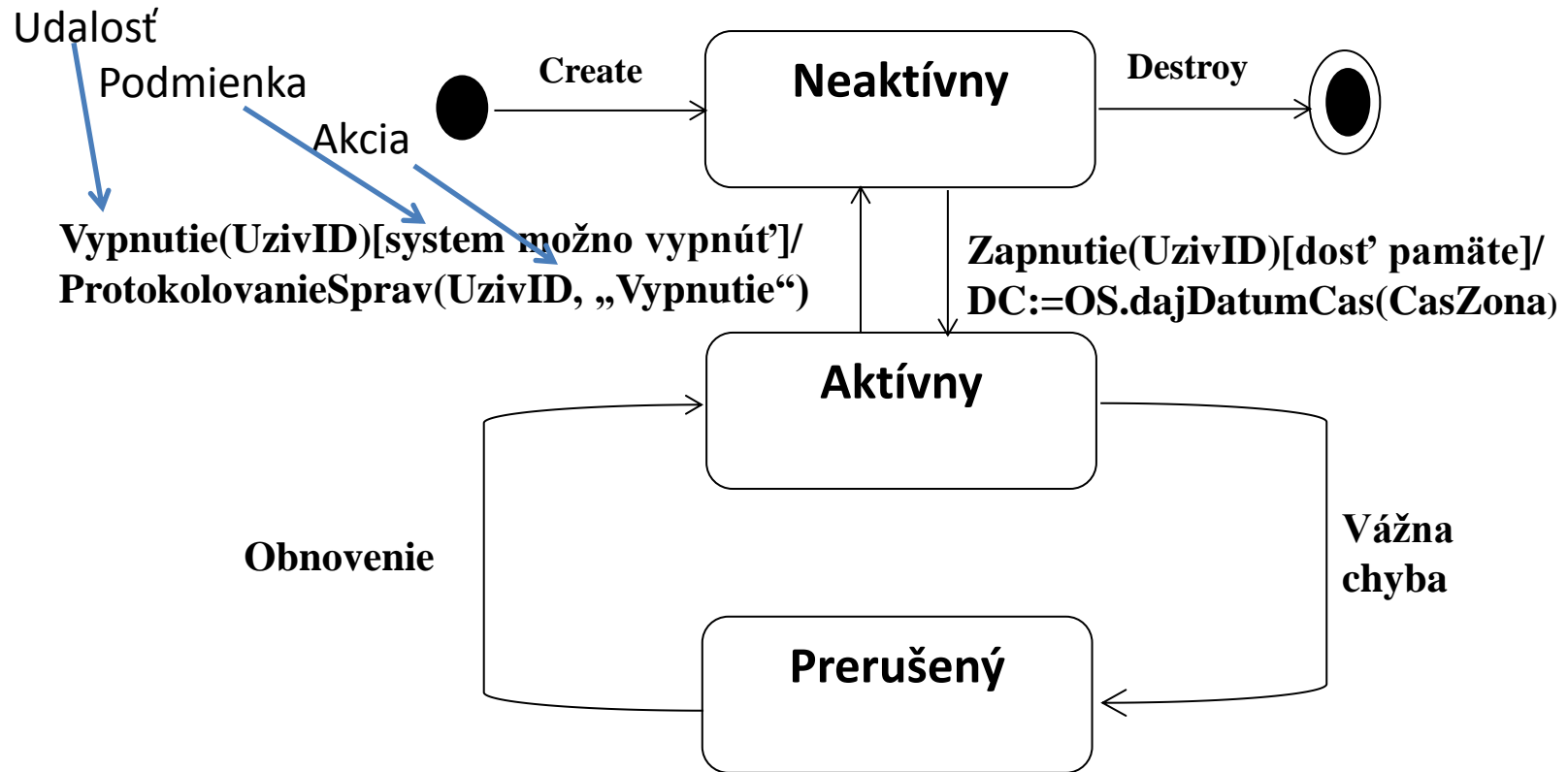
Objekty sú vždy v nejakom stave



Stavový diagram - prechody

- rôzne stavy jedného prvku
- vzťahy medzi stavmi = prechody medzi stavmi
 1. Prvok je v východnom stave
 2. Nastane udalosť
 3. Vykoná sa akcia
 4. Prvok je v cieľovom stave
- automatický prechod – bez popisu
- prechod s popisom
 - UML:
 - plná čiara od východzieho stavu k cieľovému
 - meno:
udalosť'/akcia

Stavový diagram - príklad



Sekvenčný diagram - Sequence diagram

- informácie o vzájomnej reakcii a pôsobení objektov na seba v čase
- čiarý život** - reprezentuje existenciu prvku v čase
- aktivácie** - reprezentujú časový úsek, periódu počas ktorej prvok vykonáva operáciu
- komunikácie** - horizontálna plná čiara so šípkou od čiarý života alebo aktivácie odosielateľa k čiare života alebo aktivácii prijímateľa

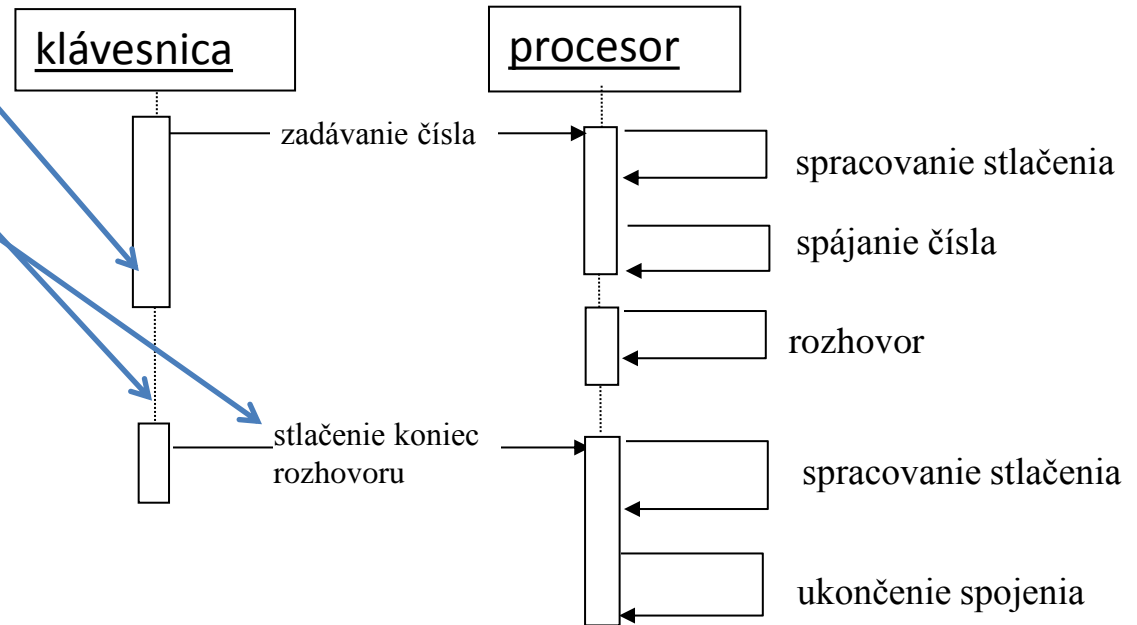


Diagram komponentov – Component diag.

- **Komponent** = časť systému, ktorá existuje, keď je systém „vykonávaný“ (keď program beží)

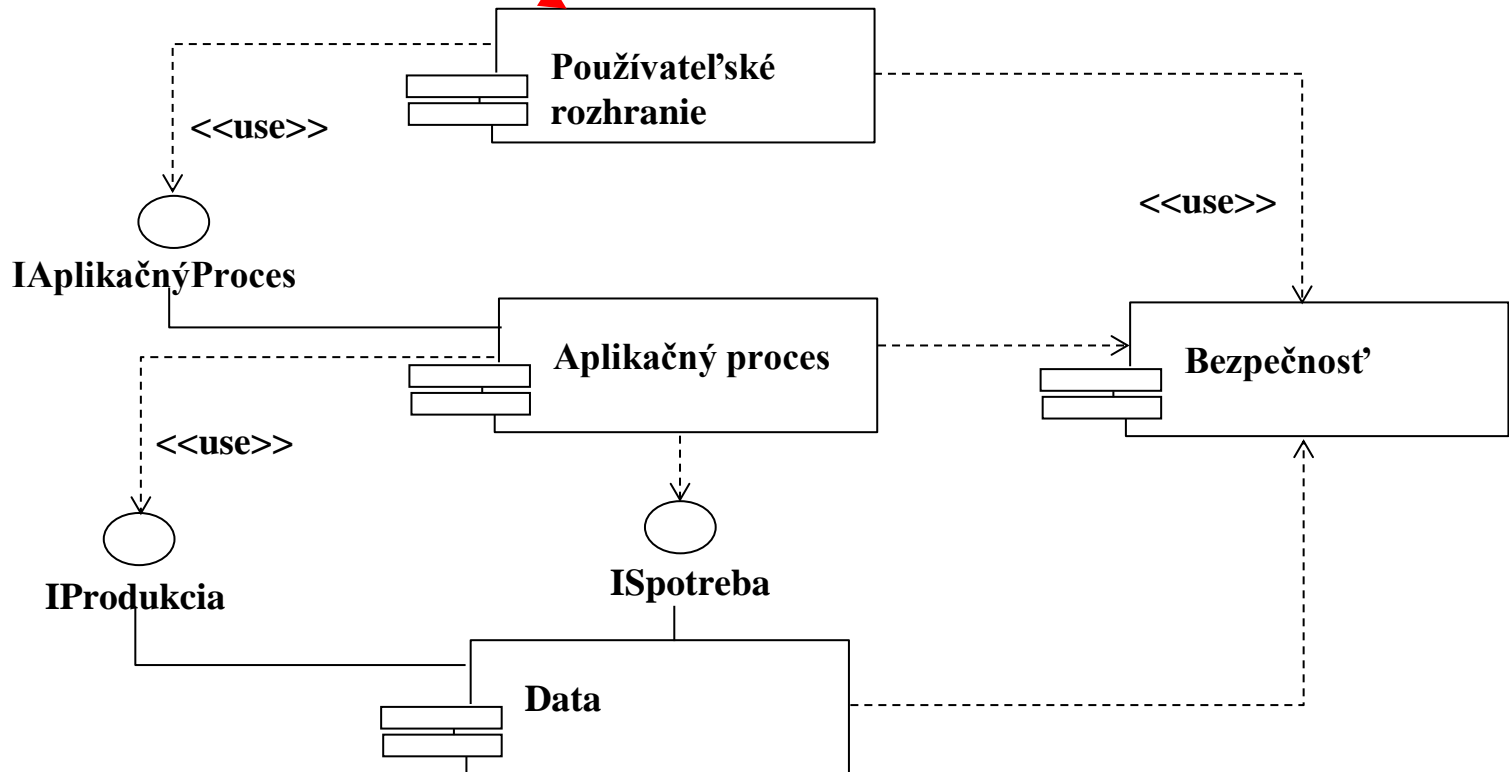
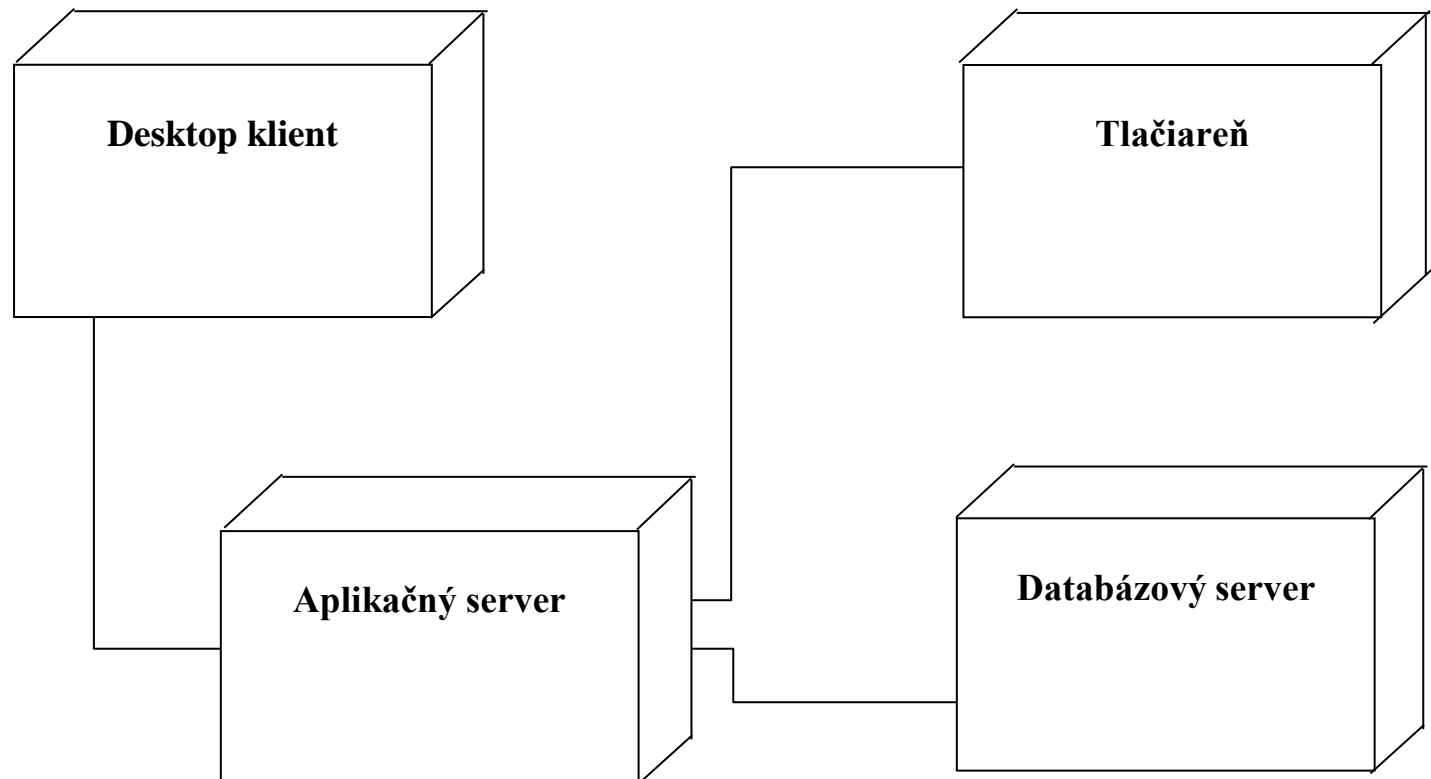


Diagram nasadenia – deployment diagram

- Ukazuje spojenie časti systému
- Prvky = uzly
- Vzťahy = asociácie



- UML – čo to je
- Diagramy
- Príklady

Ďakujem za pozornosť

Vaše otázky...

