

PB007 Softwarové inženýrství I

Cvičení 2 – organizácia testíkov, úvodný Use Case Diagram

Valdemar Švábenský

Fakulta informatiky, Masarykova univerzita, Brno

29. septembra 2015



- 1 Organizačné pokyny k testom na cvičení
- 2 Požiadavky, Use Case Diagram
- 3 Úlohy
- 4 Visual Paradigm: tipy

① Organizačné pokyny k testom na cvičení

② Požiadavky, Use Case Diagram

③ Úlohy

④ Visual Paradigm: tipy

Testík na cvičení

- Elektronický odpovedník v ISe
- 10 odpovedníkov: od 3. do 12. cvičenia (vrátane)
 - Skúšobný, nebodovaný odpovedník na 2. cvičení
- 4 otázky po 0,5 b k UML diagramom z prednášok
 - 0–2 b na jedno cvičenie, až 20 b za celý semester
 - Žiadne záporné body za chybné odpovede
 - –2 b za odpisovanie
 - Jediná správna odpoveď (radio button) alebo viac správnych odpovedí (checkbox) – musia byť všetky správne, inak 0 b
- Čas na vyplnenie: 3 minúty

Testík na cvičení

- **Jediný pokus**
- **Otvárajte len na cvičení na pokyn cvičiaceho**
- Nedodržanie pokynov (skoršie / neskoršie otvorenie odpovedníku, otvorenie mimo učebňu. . .) ⇒ bodová penalizácia a podnet k disciplinárnemu konaniu
- Ospravedlnená absencia ⇒ náhradný odpovedník na konci semestru (výber zo všetkých otázok)
- Notebookové skupiny: je možné skladať testík aj na mobile alebo tablete (na vlastné riziko)
 - Môžete si posúvať notebook v skupine, ale ostatní sa nemôžu dívať – test nie je tímová práca

① Organizačné pokyny k testom na cvičení

② Požiadavky, Use Case Diagram

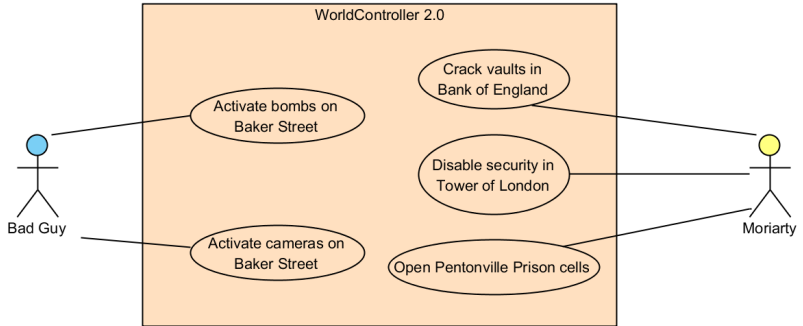
③ Úlohy

④ Visual Paradigm: tipy

Funkčné a nefunkčné požiadavky

- Požiadavky = popis systémových služieb a obmedzení
- Funkčné požiadavky = popis **služieb**, ktoré má systém poskytovať
 - Reakcie systému na isté situácie a vstupy
 - To, **čo** má systém robiť (alebo nerobiť)
 - Napr. „Systém umožňuje učiteľovi vytvoriť kurz“
- Nefunkčné požiadavky = popis vlastností systému a **obmedzení** na systém (spoľahlivosť, bezpečnosť, ...)
 - **Ako** má systém niečo robiť
 - Napr. „Systém ukladá údaje o kurzoch v DBMS Oracle“
 - Buďte špecifickí

Use Case Diagram – ukážka (Sherlock)

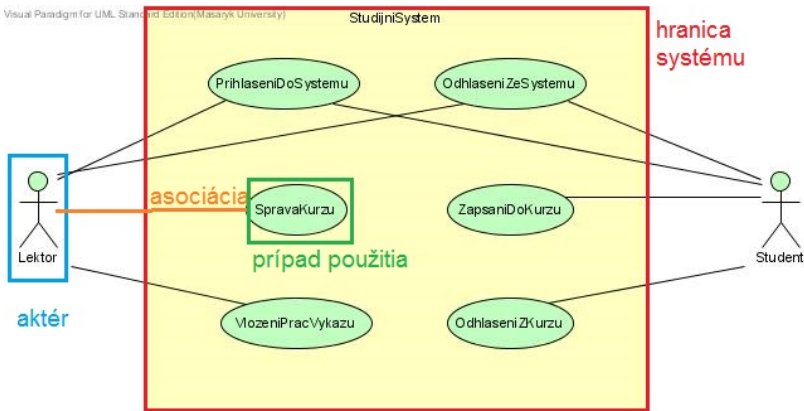


Use Case Diagram

- Grafické vyjadrenie **funkčných požiadaviek** na systém
- Definuje **vonkajší pohľad** na systém
- Tvorí ho:
 - Hranica systému (System boundary / Subject)
 - Aktéri (Actors) = role entít externých k systému, ktoré priamo interagujú so systémom
 - Prípady použitia (Use cases) = funkcionality požadované (a spúšťané) aktérmi
 - Vzťahy / asociácie (Relationships) medzi aktérmi a prípadmi použitia
- Zachytáva komunikáciu medzi aktérmi a systémom

Use Case Diagram – ukážka

Visual Paradigm for UML Standard Edition (Masaryk University)



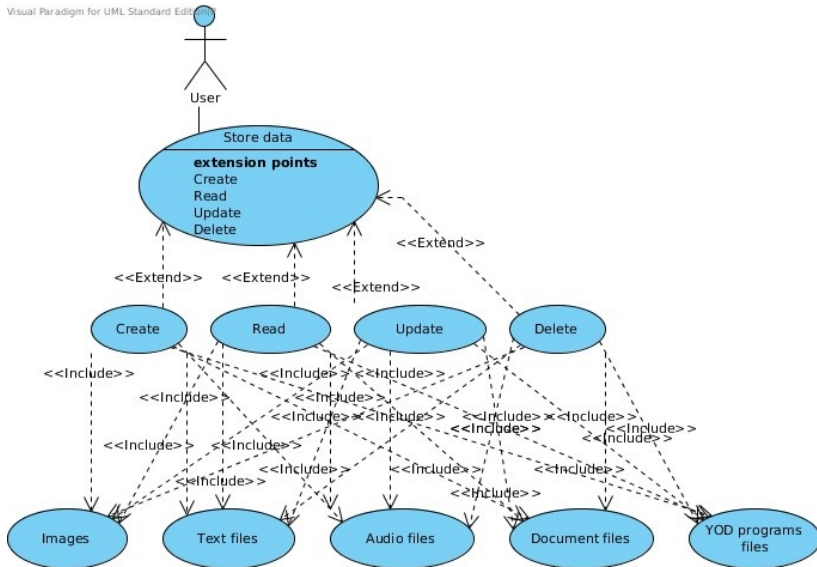
Zdroj: https://is.muni.cz/auth/el/1433/podzim2015/PB007/um/sem/cz_files/02/02_Studium_UseCase.jpg

Postup pri modelovaní

- 1 Vymedzenie hraníc systému
- 2 **Nájdenie aktérov**
- 3 **Nájdenie prípadov použitia**
- 4 Určenie vzťahov medzi aktérmi a prípadmi použitia
- 5 Špecifikácia prípadov použitia

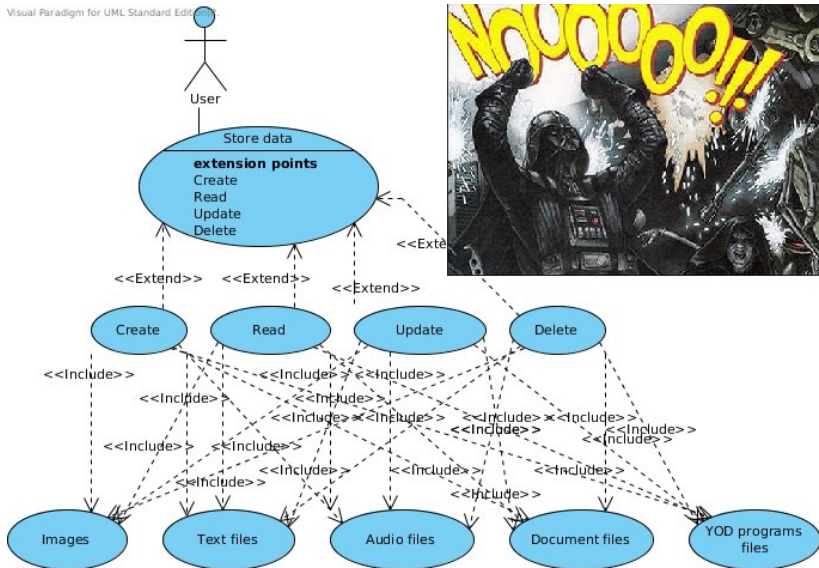
“Keep It Simple, Stupid”

Vytvorte prehľadný Use Case



Vytvorte prehľadný Use Case

Visual Paradigm for UML Standard Edition



Identifikácia aktérov

- Kto alebo čo používa daný systém?
- Akú rolu zohráva pri tejto interakcii?
- Aké ďalšie systémy spolupracujú s našim systémom?
- Kto alebo čo získava informácie zo systému?
- Kto alebo čo zadáva informácie do systému?
- Dochádza k nejakej udalosti pravidelne alebo v pevne danom čase?
- Používajte jasné a jedinečné označenie podstatným menom v jednotnom čísle

Identifikácia prípadov použitia

- Prípad použitia začína akciou tzv. **primárneho** aktéra
- Prípad použitia môže ovplyvniť tzv. **sekundárnych** aktérov
- Aké funkcie požaduje konkrétny aktér od systému?
- Ukladá a získava systém nejaké informácie? Ak áno, ktorí aktéri spúšťajú tieto činnosti?
- Čo sa stane pri zmene stavu systému? Sú o tom aktéri informovaní?
- Existujú externé udalosti, ktoré ovplyvňujú systém? Čo upozorní systém na tieto udalosti?
- Používajte jasné a jedinečné označenie v tvare slovesnej väzby z pohľadu aktéra

① Organizačné pokyny k testom na cvičení

② Požiadavky, Use Case Diagram

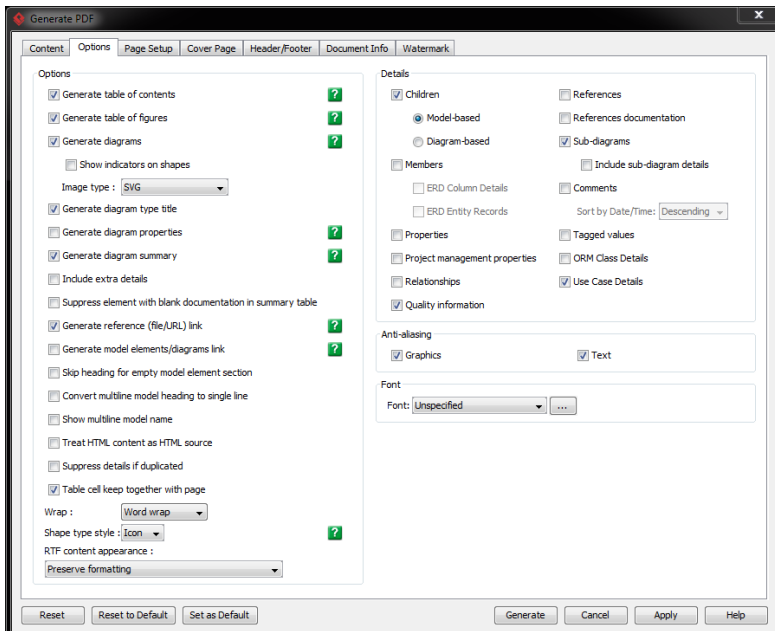
③ Úlohy

④ Visual Paradigm: tipy

Úlohy

- Do špecifikácie prázdneho diagramu z 1. cvičenia doplňte funkčné požiadavky rozdelené podľa rolí aktérov
 - Ak ste nestihli / menili od 1. cvičenia
- Spíšte tiež **nefunkčné požiadavky** (platforma, šifrovanie, zálohovanie, očakávaný počet požiadaviek za minútu, ...)
- Na základe tejto špecifikácie vytvorte Use Case Diagram (⇒ aktéri, prípady použitia a väzby medzi nimi)
 - Funkčné aj nefunkčné požiadavky budú v textovej forme v Open Specification diagramu (doplňte z 1. cvičenia)
- Vygenerujte **PDF** report a vložte ho do odovzdávarne „Week 02“ (skupiny 09, 10)
 - Názov v tvare *priezvisko1-priezvisko2-priezvisko3.pdf*
 - Neodovzdávajte súbor s projektom (.vpp)!
 - Odovzdáva jeden človek za svoj tím
 - Deadline: 4.10. 23:59

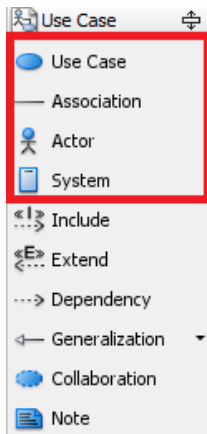
Nastavenie (Tools → Report → Generate PDF...)



- 1 Organizačné pokyny k testom na cvičení
- 2 Požiadavky, Use Case Diagram
- 3 Úlohy
- 4 Visual Paradigm: typy

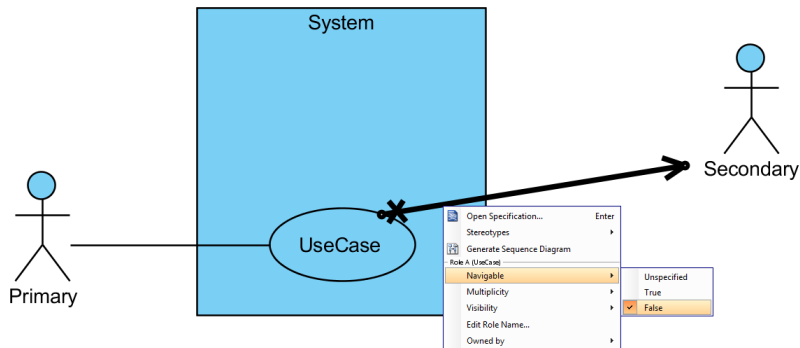
Panel nástrojov

- Stačí, keď budete používať prvé 4 položky
- Zvyšné hlavne na ďalšom cvičení



Sekundárny aktér (Supporting Actor)

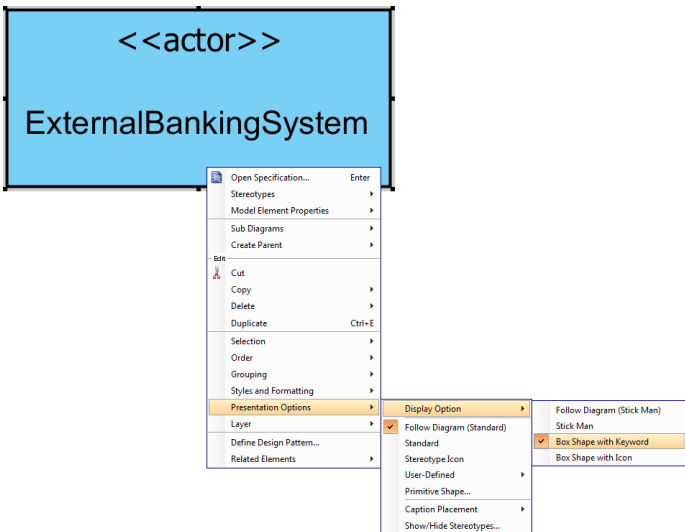
- Ak vieme zo vzťahu vyvodit jednosmernost komunikacie, smeruje k nemu jednosmerná šípka
 - Prípád použitia nespúšťa, ale je ním ovplyvnený



- Jednosmerná šípka \Rightarrow sekundárny aktér
- Ale sekundárny aktér \nRightarrow jednosmerná šípka
 - Môže zo systémom komunikovať aj obojstranne
 - Sekundárny aktér sa nedá vždy rozpoznať „na pohľad“

„Neludský“ aktér

- Aktér nemusí byť len rola, ktorú zastáva človek
- Externý systém, proces, čas, . . .



Premenovanie nadpisu PDF reportu

- Zmenili sa vám členovia tímu po vytvorení projektu?

