

Use Case Diagram

PB007 Softwarové inženýrství I

Martin Jakubička,
1.10.2012

Use Case diagram

- K čemu diagram slouží:
 - Zachycení potřeb a požadavků uživatelů na systém
 - Efektivní mechanismus pro komunikaci mezi uživateli a vývojáři
- Co přináší
 - Grafické rozhraní je dostatečně čitelné i pro laika
 - Prostředek k vyjádření konkrétních cílů, které chce uživatel dosáhnout, prostřednictvím sekvence interakcí se systémem
 - Dokumentace dostatečně popisuje konkrétní interakce

Komponenty diagramu

- Diagram se skládá z těchto částí
 - **Aktér** (Actor): uživatelská role nebo systém
 - **Hranice systému** (System boundary): vymezení hranice systému
 - **Případ užití** (Use Case): dokumentace události, na kterou systém reaguje
 - Vazby mezi aktéry a případy užití

Aktér

- Vyjadřuje **roli**, ne konkrétní osobu
- Role může být osoba nebo předmět používající daný systém (lidé, jiné systémy, čas)
- Interakce v modelu probíhá vždy mezi aktérem a systémem, ne mezi aktéry
- Názvy aktérů by měly být krátké a výstižné
- Pro více informací je vhodné doplnit popis

Aktér

- Při hledání aktérů si položíme otázky:
 - Kdo do systému přistupuje?
 - Jakou roli zastává?
 - Jaké další systémy komunikují s modelovaným systémem?
 - Vystupuje v systému nějaká časová událost?

Hranice systému

- Definice hranice systému probíhá interaktivně
- Hranice mohou být definovány na různých stupních abstrakce
 - Vysoká: bez reprezentantů
 - Střední: kdo co skutečně vkládá
 - Nízká: jaká data skutečně vstupují

Případ užití

- Funkce, které systém poskytuje aktérům
- Každý případ užití musí být dokumentovaný
- Dokumentace je psaná z pohledu aktéra
- Pojmenování musí vyjadřovat danou akci (tj. nepoužívat např. personalistika, sklad,... ale vytvořitObjednávku, zálohovatSystém, apod.)

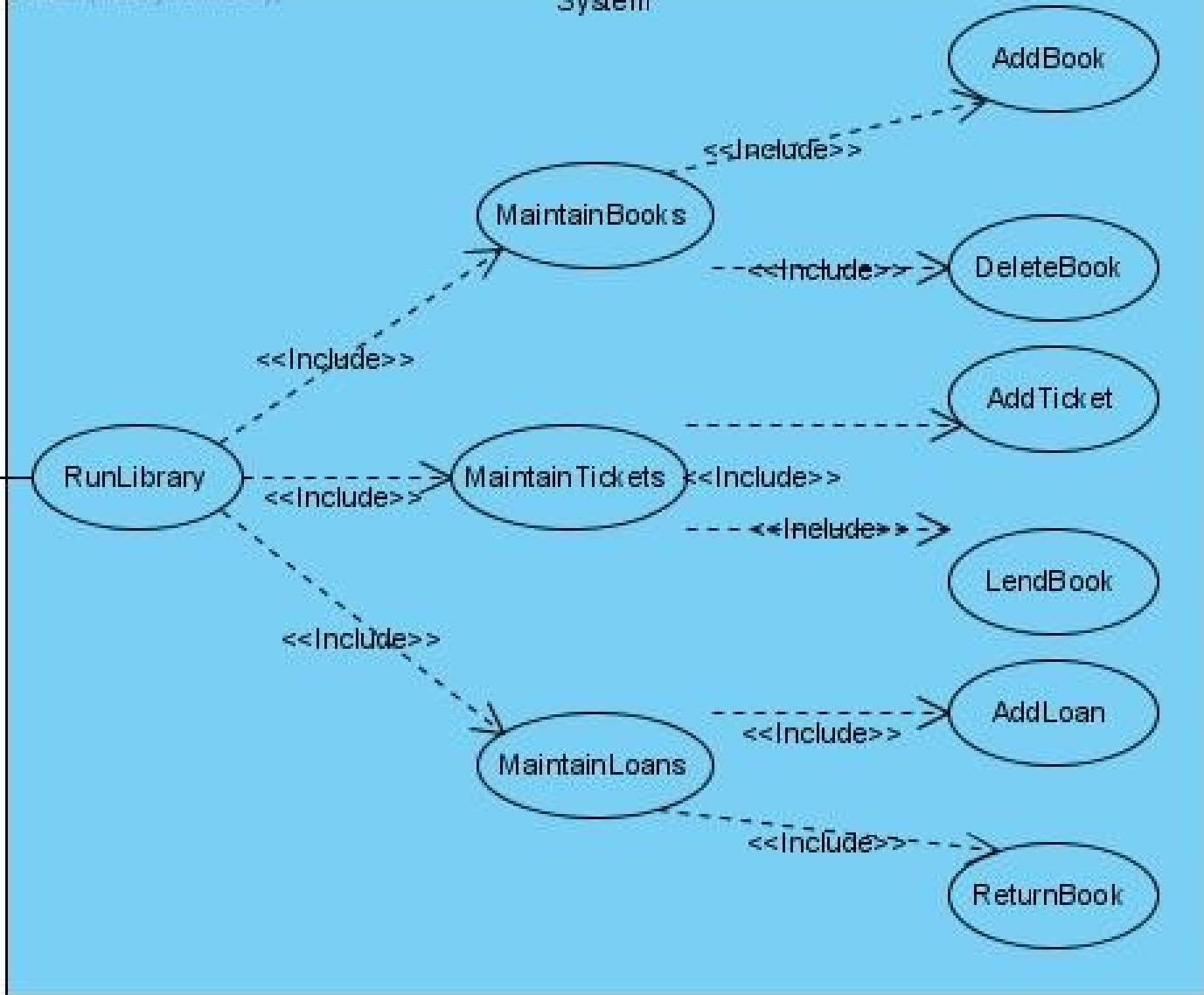
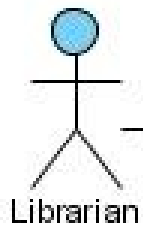
Případ užití

- Při hledání případu užití si položíme otázky:
 - Jaké funkce aktéři očekávají od systému?
 - Bude systém poskytovat a uchovávat nějaké informace? Pokud ano, mohou aktéři s těmito informacemi manipulovat pomocí akcí?
 - Existují vnější události vstupující do systému?
 - Jací aktéři budou upozorněni na změnu stavu systému?

Rada na závěr

- Vyhněte se funkční dekompozici
- Nadbytečné propojování případů užití vede k nepřehlednosti modelu
- Ukázka jak to NEMÁ vypadat....

System



Úkoly

- Založte nový projekt ve Visual Paradigm:
File → New Project, UML verze 2.1, název projektu:
příjmení všech řešitelů bez diakritiky oddělených
pomlčkou, popis: zadání projektu
- Navrhněte prvotní model případu užití – nalezení
hranice systému, aktérů, případů užití a vazeb mezi
aktéry a případy užití
- Hledejte mezery ve specifikaci a navrhněte pro tyto
mezery řešení, které pak konzultujte se cvičícím
- Vložte poznámku (Note) kde budou navržené doplňující
informace ke specifikaci