

OOT

Objektově orientované technologie

Požadavky a případy užití

Daniela Szturcová, Pavel Děrgel
Institut geoinformatiky, HGF

Osnova

- Systém
- Uživatelé
- Případy užití
- Vazby (asociace, generalizace, include a extend)
- Shrnutí

Modelování případů užití

- Jedná se o způsob zachycení případů, kdy systém používá některý uživatel.
- Cílem je:
 - Nalezení hranic systému.
 - Nalezení uživatelů.
 - Nalezení případů užití.
- Výsledkem je specifikace toho co systém umí (bude umět) a seznam uživatelů, kteří tyto funkce využívají.

Co jsou případy užití

- Mají ukázat, jaké jsou typické interakce mezi uživateli (kdo/co) a systémem.
- Jedná se o příběhy užití systému.

Co jsou případy užití

- Ukazují kdo a jak bude systém používat.
- Jaké funkce by měl nabízet.
- Jací uživatelé s ním budou pracovat.
- Zachycuje systém z pohledu zákazníka a jeho požadavků.
- Lze je aplikovat na nejrůznější typy systémů (nejen softwarové).

Co jsou případy užití - scénář

- Zákazník přistoupí k nápojovému automatu, zvolí si požadovaný nápoj, vhodí mince do otvoru.
- Automat zkontroluje obnos.
- Nápoj vypadne ze zásobníku, zákazník si nápoje odebere.
- Co když obnos nestačí?
- Co když došly nápoje?
- Jaký je cíl zákazníka?

Scénář případu užití

UC:PlatitDaňZPřidanéHodnoty

ID: UC1

Stručný popis:

Na konci fiskálního čtvrtletí zaplatit daň finančnímu úřadu.

Hlavní aktéři:

Čas

Vedlejší aktéři:

Finanční úřad

Vstupní podmínky:

1. Je konec fiskálního čtvrtletí?

Hlavní scénář:

1. UC začíná, je-li konec fiskálního čtvrtletí.
2. Systém zjistí částku, kterou je třeba zaplatit finančnímu úřadu.
3. Systém odesílá elektronickou platbu finančnímu úřadu.

Následné podmínky:

1. Finanční úřad přijímá DPH ve správné výši.

Alternativní scénáře:

Scénář případu užití

- Hlavní užitečný scénář.
- Postupné kroky k dosažení cíle.
- Diskuze k alternativním případům:
 - Mohla by nastat změna v nějakém kroku?
 - Mohlo by se něco pokazit?
- Rozšíření, vložení alternativ.

Proč jsou případy užití důležité?

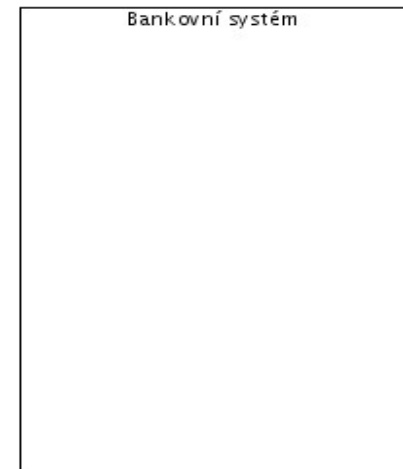
- Nástroj pro komunikaci s uživateli.
- Cílem je zapojit budoucí uživatele do analýzy a návrhu systému.
- Zvyšuje úspěšnost tvorby systému.

Základní pojmy

- **Uživatel** (Aktor) – člověk nebo jiný systém, který vystupuje v určité *roli* a interaguje se systémem.
- **Případ užití** (UseCase) – popisuje činnost (nebo množinu činností), které poskytují uživateli nějakou službu.
- **Vazby** – popisují vztah mezi uživatelem a případem užití, nebo mezi dvěma uživateli (popř. dvěma případy užití).
- **Hranice systému** – definuje co je a co není součástí systému.

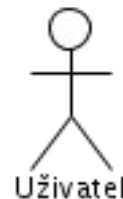
Hranice systému

- Definuje co je a co není součástí systému
- je znázorněna např. jako rámeček s názvem systému
- účastníci stojí mimo „rámeček“
- případy užití stojí uvnitř



Nalezení účastníků

- Účastník specifikuje roli, kterou určitá externí entita přijímá v okamžiku, kdy začíná daný systém využívat.
- Jsou to **role**, které hrají externí subjekty při používání systému:
 - lidé,
 - jiné systémy,
 - čas.



Účastníci (aktéři)

- Účastníci jsou externí subjekty, ale systém může udržovat určitou interní reprezentaci některých účastníků (identifikace, přístup do systému ...).
- Účastníkem může být i **čas** (tam, kde některé činnosti probíhají automaticky v určitém čase).

Vyhledání účastníků

- Zvažujeme:
 - Kdo nebo co daný systém používá?
 - Jaká je role při komunikaci se systémem?
- Pomohou odpovědi na otázky:
 - Kdo nebo co používá daný systém?
 - Jakou roli v této interakci hraje?
 - Kdo instaluje systém?
 - Kdo spouští a vypíná systém?
 - Jaké další systémy spolupracují se systémem?
 - Děje se něco v určitou dobu?

Vyhledání účastníků

Je nezbytné pamatovat na tyto body:

- účastníci stojí mimo systém - nejsou tedy pod naší kontrolou,
- účastníci komunikují přímo se systémem - to napomáhá stanovení hranice systému,
- účastníci představují role hrané externími subjekty, nejsou to tedy přímo tyto subjekty,
- jeden externí subjekt může hrát i více rolí.

Vyhledání účastníků

- Každý účastník musí mít krátký výstižný název, srozumitelný všem.
- Každý účastník musí mít krátký výstižný popis (jeden až dva řádky), popisující jeho roli při užívání systému.

Účastníci (aktéři)

Aktér	Význam
Zákazník	Kdokoli, kdo si rezervuje či zakoupí jízdu taxíkem od fy XY.
Dispečer	Zaměstnanec – operátor, který je zodpovědný za evidenci údajů o rezervacích, zaměstnancích a vozidlech.
Administrátor systému	Speciální uživatel systému, který spravuje systém TaxiS (práva ostatních uživatelů, zavedení nového uživatele, ap.).
Zaměstnanec	Zaměstnanec podílející se na činnostech spojených s taxislužbou a fungováním systému TaxiS.
Řidič	Zaměstnanec - řidič, který se zákazníkem a vozidlem vykoná jízdu.

Co je to UseCase?

- UseCase je funkce systému, kterou uživatel požaduje (případ užití).
- UseCase je vždy spuštěn uživatelem:
 - primární uživatel spouští UseCase,
 - 0 nebo více sekundárních uživatelů se mohou účastnit jednoho případu užití.
- UseCase se vždy zapisují z pohledu uživatele.

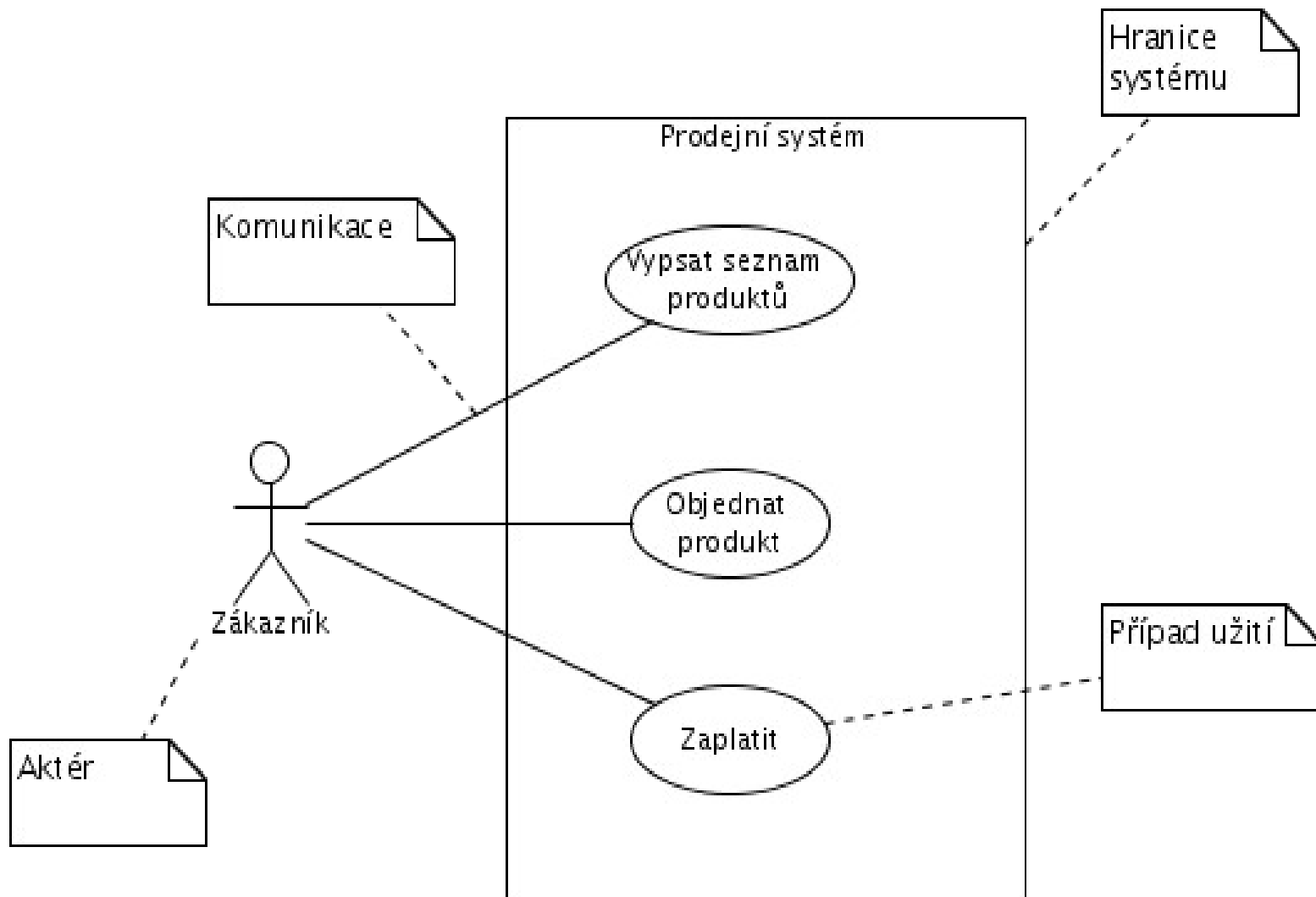
Objednat

ZjistitStav
Objednávky

Hledání UseCase

- Začátek - seznam uživatelů, kteří budou se systémem pracovat.
- Pomohou další otázky:
 - Jaké funkce jednotliví uživatelé od systému očekávají?
 - Co se stane, když se změní stav systému? Jsou o tom uživatelé informováni?
 - Existují nějaké vnější události, které ovlivňují systém?
 - Komunikuje systémem s jiným systémem?
 - Produkuje systém nějaké výstupy?

Diagramy případů užití



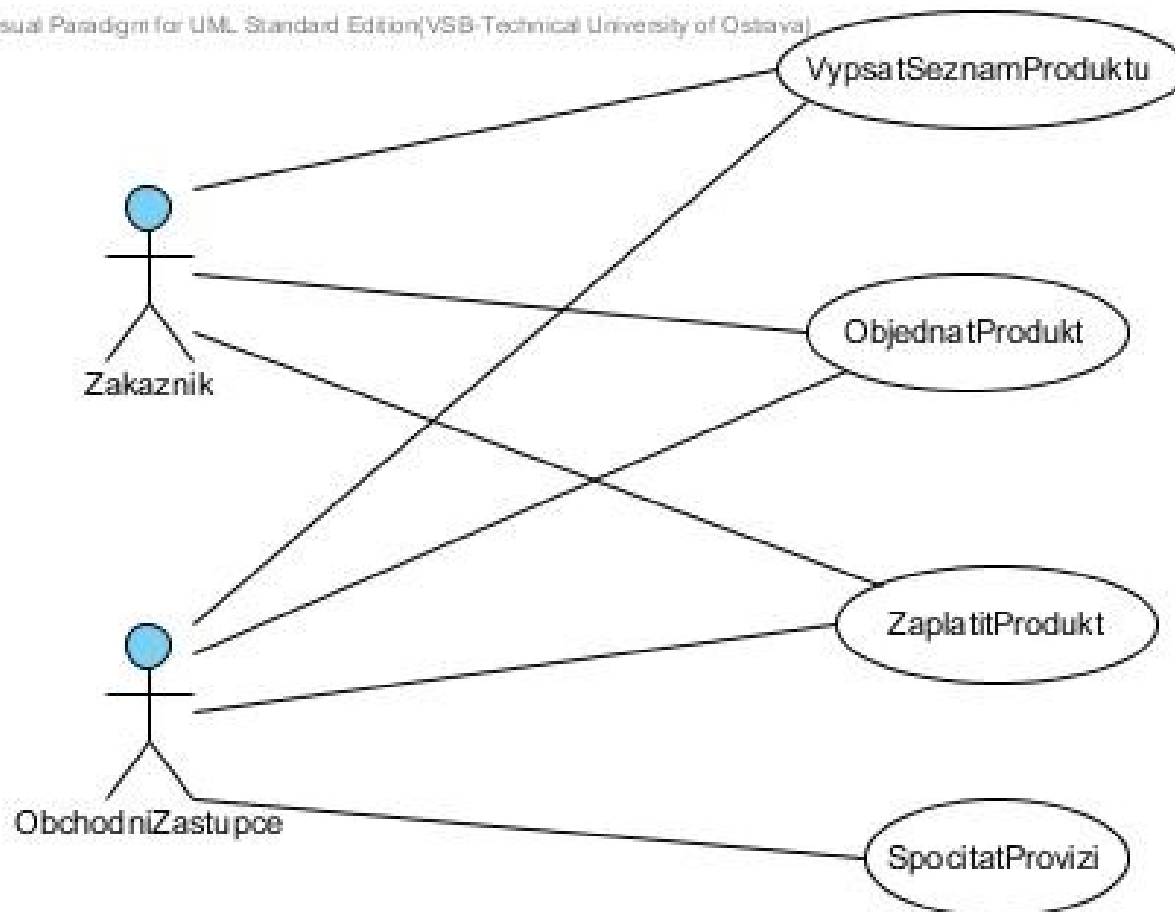
Zobecnění aktérů

- Potomka můžeme dosadit všude tam, kde lze očekávat jeho předka.

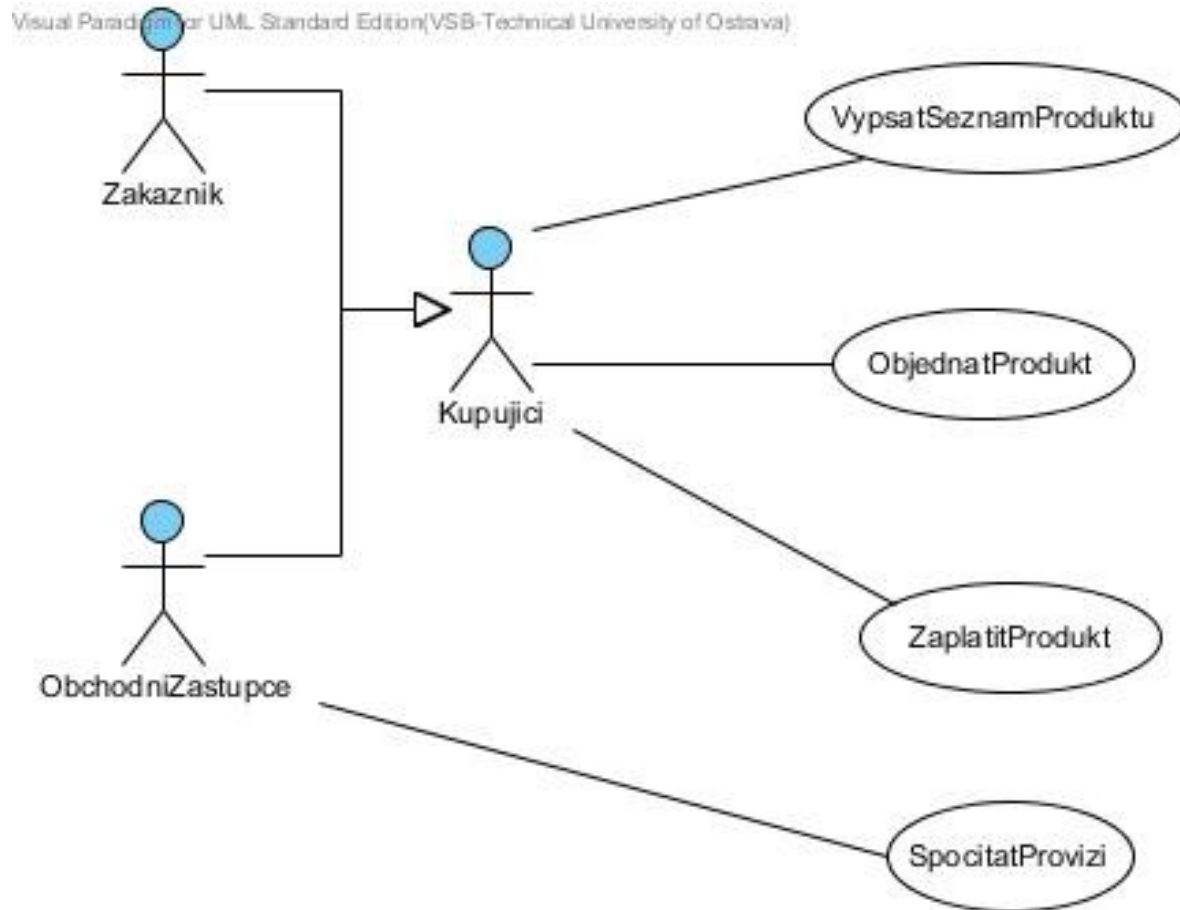
Zobecnění aktérů

- Potomek přebírá od předka.

Visual Paradigm for UML Standard Edition (VSB-Technical University of Ostrava)



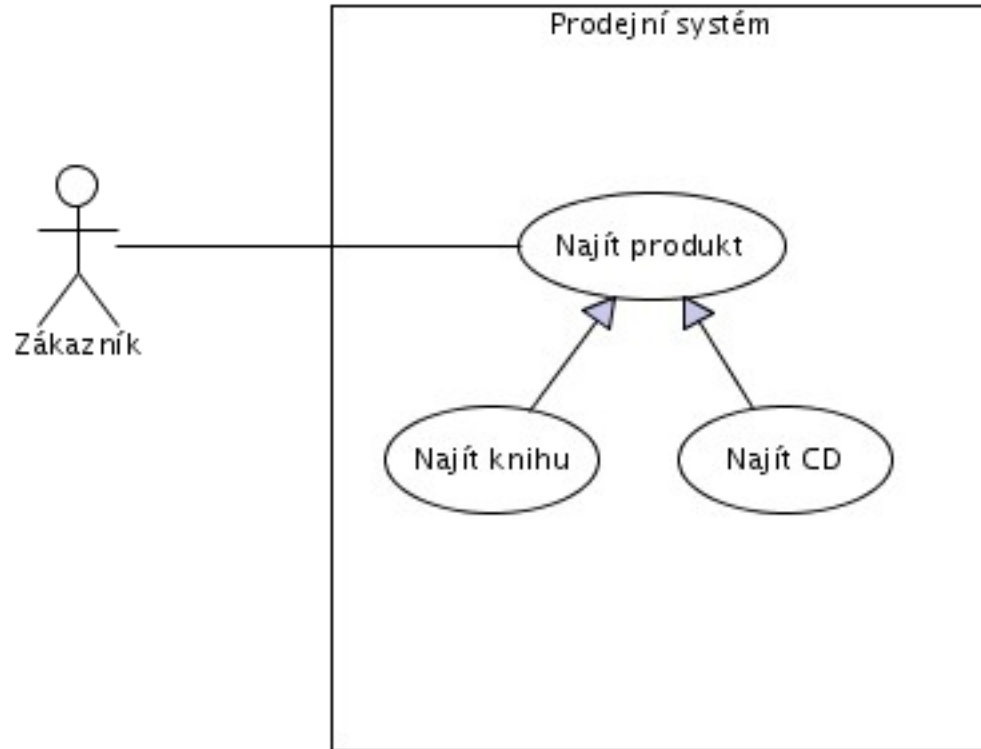
Zobecnění aktérů



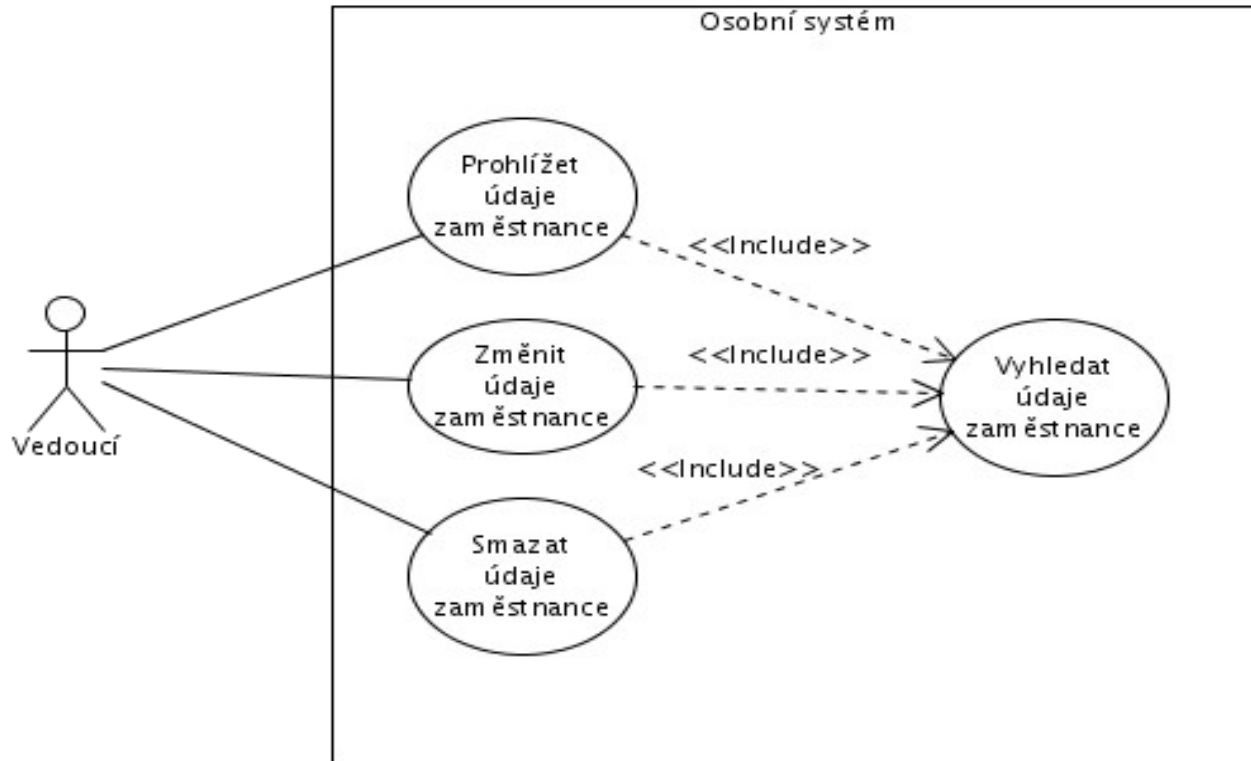
Zobecnění případů užití

- Odvozené případy užití mohou:
 - dědit funkce a vlastnosti od svých předků,
 - přidávat nové funkce a vlastnosti,
 - překrývat (měnit definici) zděděných funkcí a vlastností.
- Odvozený případ užití automaticky dědí všechny funkce a vlastnosti předka.

Zobecnění případů užití

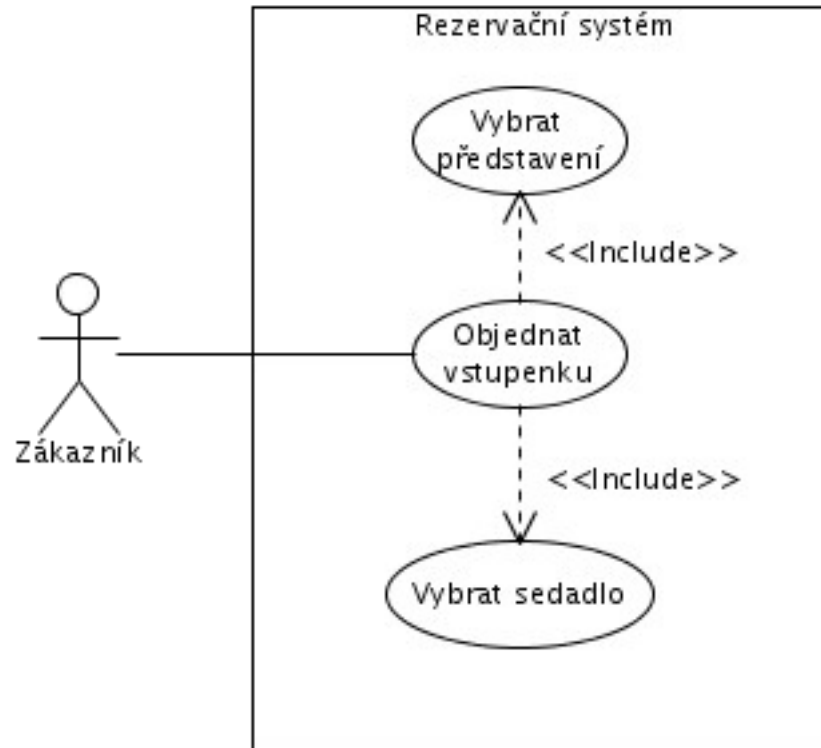


Relace «include» (vložení)

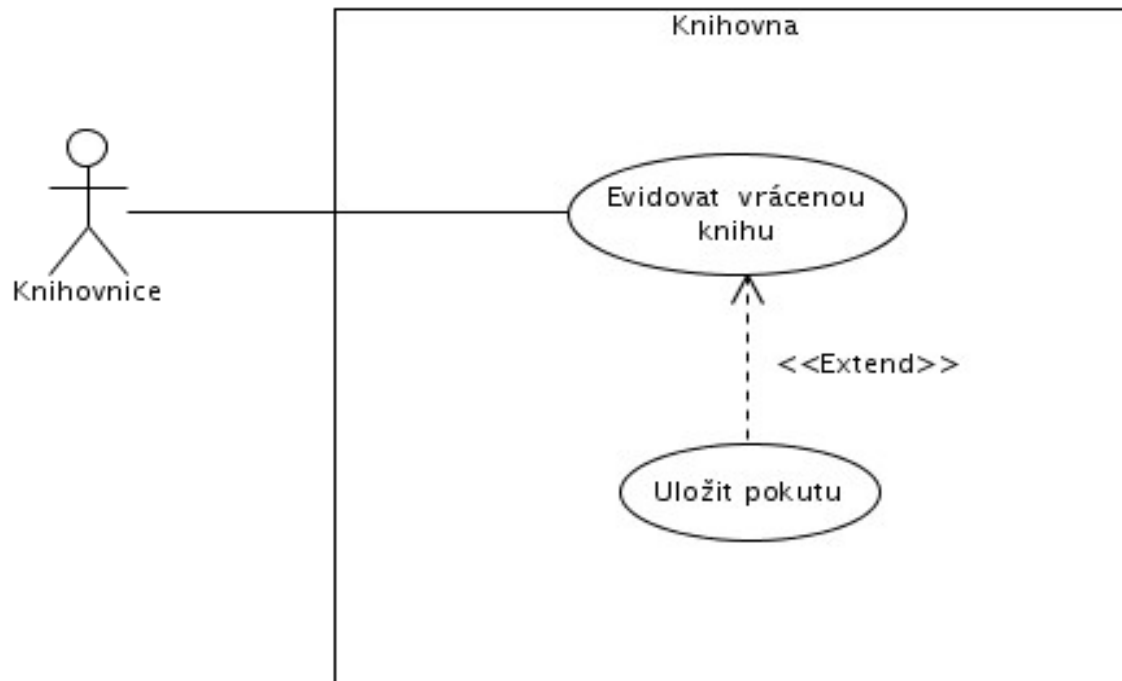


- Vkládaný prvek je nutný pro existenci vkládajícího.
- Vkládaný prvek je obvykle opakující se rutina.

Relace «include»



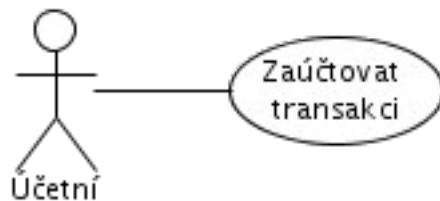
Relace «extend» (rozšíření)



- Vkládaný prvek je volitelný.
- Doplnuje (rozšiřuje) chování základního případu užití.

Model jednání

- Model jednání se skládá:
 - diagram případů užití,
 - slovní scénář pro každý případ užití,
 - podmínky.

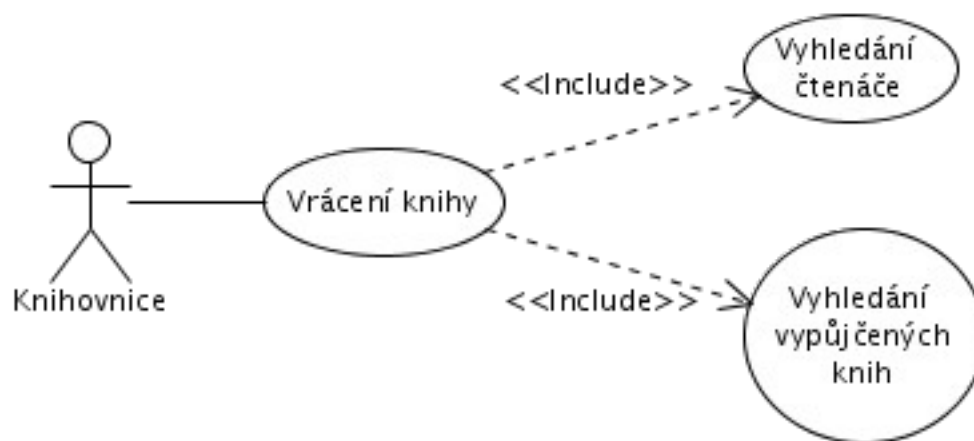


1. Otevření aplikace
2. Výběr typu transakce
3. Volba účtu
4. Vložení popisu transakce, částky
5. Potvrzení správnosti
6. Zaúčtování

Model jednání

- Zabývá se podrobnější specifikací identifikovaných případů užití.
- Výstupem jsou podrobnější případy užití.
- Skládá se alespoň z názvu a specifikace případu užití.
- Může (nepovinně) obsahovat i stručný popis.

Model jednání (formálně)



Super Use Case			
Author	Pavel Děrgel		
Date	1/08/2007 11:59:04		
Brief Description	Vrácení knihy		
Preconditions	1. čtenář chce vrátit vypůjčenou knihu 2. Knihovnice je přihlášená do systému		
Post-conditions	Seznam knih je aktualizován		
Flow of Events		Actor Input	System Response
	1		Nalezení čtenáře
	2		Nalezení vypůjčených knih
	3	Knihovnice vybere knihy, které chce čtenář vrátit	
	4	Potvrzení výběru	
5		Zaznamenání vrácení knih	

Scénář případu užití

UC: RealizovatJizdu

ID: UC4

Stručný popis: Zakaznik nastoupí do vozidla na místě nástupu, v cíli jízdy mu Ridic vypočte cenu za jízdu, Zakaznik zaplatí.

Hlavní aktéři: Zakaznik

Vedlejší aktéři: Ridic

Vstupní podmínky: Ridic s Vozidlem je na místě nástupu. Zakaznik požaduje odvézt z místa nástupu do cíle.

Hlavní scénář:

1. Zakaznik na místě nástupu nastoupí do připraveného vozidla.
2. Ridic nastaví sazbu dle nejvhodnějších podmínek pro Zakaznika.
3. Ridic jede po optimální trase do cíle zadaného Zakaznikem.
4. V cíli Ridic vypočte cenu za Jizdu.
5. Zakaznik zaplatí vypočtenou cenu.

Následné podmínky: Ridic s Vozidlem je na místě cíle a je uvolněn k další Jizde.

Alternativní scénáře:

K čemu je to všechno dobré?

- Případy užití jsou důležité pro
 - úvodní etapu analýzy problému,
 - návrh uživatelského rozhraní systému,
 - komunikaci se zadavatelem a budoucími uživateli (je to součást zadání),
 - vývojáře ,
 - testování výsledného systému,
 - ověření zda výsledný systém odpovídá zadání.

Kdy tvořit případy užití?

- UseCase je vhodný nástroj pokud:
 - Existuje hodně funkčních požadavků na systém.
 - Systém má mnoho uživatelů.
 - Systém má mnoho rozhraní pro komunikaci.
- UseCase se nedoporučuje pokud:
 - Systém poskytuje jednu nebo jen malý počet funkcí.
 - Se systémem pracuje jeden uživatel.
 - Systém nabízí málo rozhraní.

Shrnutí

- Specifikace požadavků vytváří představu o nárocích na systém.
- Případy užití – popisují, jak budou se systémem pracovat jeho budoucí uživatelé a jaké funkce bude systém nabízet.
- Hranice systému je jasně definována.
- Případy užití mohou být vkládány do jiných nebo rozšiřovány.

Použité zdroje

- Tom Pender.: *UML Bible*.
- Jim Arlow, Ila Neustadt.: *UML2 a unifikovaný proces vývoje aplikací*.
- *Materiály Pavla Děrgela*.

Děkuji za pozornost