

Prednáška 2: Prípady použitia

Metódy a prostriedky špecifikácie 2013/14

Valentino Vranič

Ústav informatiky a softvérového inžinierstva
Fakulta informatiky a informačných technológií
Slovenská technická univerzita v Bratislave

1. október 2013

Obsah prednášky

- 1 Čo je prípad použitia
- 2 Ako opísať prípad použitia
- 3 Diagram prípadov použitia v UML
- 4 Rozdiely v notáciách
- 5 Čo ďalej s prípadmi použitia

Čo je prípad použitia

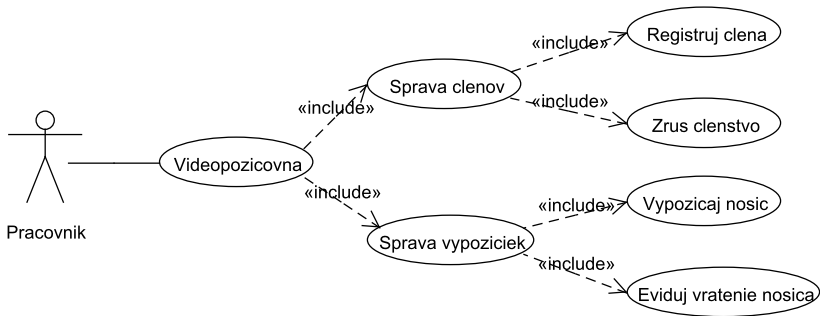
Čo je prípad použitia

- Use case – Ivar Jacobson, 1986; ale pôvodná idea z roku 1960
- „A use case models the behavior of a system. A use case is a sequence of actions performed by a system, which yields an observable result that is typically of value for one or more actors or other stakeholders of the system.“ (Jacobson)¹
 - Prípad použitia teda opisuje funkcionality systému z pohľadu používateľa
- Prípad použitia zachytáva zmluvu medzi zainteresovanými (stakeholders) v danom systéme ohľadom jeho správania (Cockburn)
- Prípad použitia opisuje súvislu funkcionality, ktorá dáva výsledok hodnotný pre používateľa (Arloow–Neustadt)

¹I. Jacobson and P. W. Ng. Aspect-Oriented Software Development with Use Cases. Addison-Wesley, 2004.

Prípady použitia a UML

- Prípady použitia nie sú nevyhnutne spojené s UML
- Prípady použitia sú staršie než UML (1986/1960 vs. 1994)
- UML je primárne grafický jazyk, kým prípady použitia sú textové opisy
- Prípady použitia pôvodne nemali grafickú podobu
- Prípady použitia boli integrované do UML len vo forme za týmto účelom vytvorených diagramov
- Diagram prípadov použitia poskytuje prehľad prípadov použitia a ich vzťahov
- Ale *prípady použitia sú primárne textové opisy*



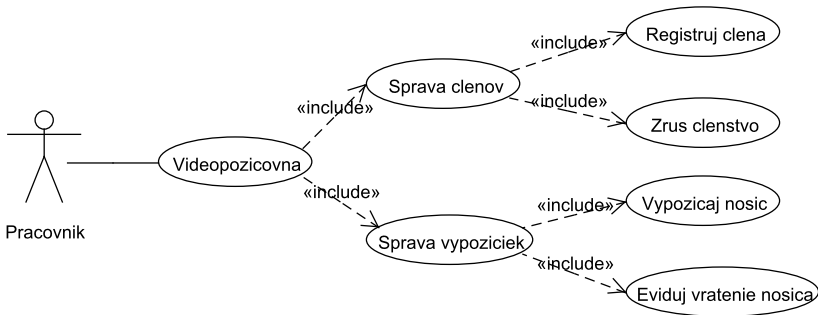
Ako identifikovať prípady použitia (1)

- Kandidátom je vždy nejaká ucelená funkcionálna, ktorá má zmysel a hodnotu pre používateľa:
 - Zapiš do ročníka
 - Vyber predmet
 - Vytvor poistnú zmluvu
 - Uprav poistnú zmluvu
 - Vybav objednávku
- Základom je zámer používateľa: čo chce dosiahnuť

Ako identifikovať prípady použitia (2)

- Prípady nevhodných „vysokoúrovňových“ prípadov použitia:
 - Správa zápisu – všetko, čo sa dá robiť so zázpismi
 - Správa poistných zmlúv
- Prípady nevhodných „nízkoúrovňových“ prípadov použitia:
 - Zadávanie mena používateľa – naozaj je to pracovný zámer používateľa?
 - Uloženie súboru – závisí od kontextu
- Vhodné priblíženie: odpoveď používateľa na otázku čo robí – alebo čo by chcel prikázať systému aby urobil
- Niektoré činnosti budú zložitejšie, iné jednoduchšie – stále to môžu byť prípady použitia
- Ale prípady použitia nepredstavujú funkcionálnu dekompozíciu systému

Prípady použitia nepredstavujú funkcionálnu dekompozíciu systému



Zlý model!

Zdroje prípadov použitia

- Hlavným zdrojom je špecifikácia požiadaviek
- Biznis model je ďalším možným zdrojom
- Často vzniká v bezprostrednej spolupráci so zákazníkom
- Účelom špecifikácie požiadaviek a modelu prípadov použitia je zistiť čo všetko systém má obsahovať
 - Ale v praxi je to inak: zistiť čo systém už nemá obsahovať (a dať si to potvrdiť u zákazníka)
 - Preto je dôležité, aby všetci zainteresovaní – vrátane zákazníka – vedeli čítať prípady použitia

Ako opísať prípad použitia

Opis prípadu použitia

- Opis prípadu použitia je primárne text
- Rôzne úrovne detailu – podľa potrieb projektu
- Model prípadov použitia nie je opis (grafického) používateľského rozhrania (GUI)
 - Prípad použitia nie je formulár GUI – ani sa nemusí viazať na jeden formulár
 - Ani sa v opisoch prípadov použitia nemajú spomínať prvky GUI
 - Prípady použitia v značnej miere určujú vhodné GUI

Základné prvky opisu prípadu použitia

- Názov
- ID – voliteľné
- Predpoklady a dôsledky
- Účastníci:
 - primárni – aktivujú prípad použitia
 - sekundárni
- Krátky opis
- Základný tok/toky
- Alternatívne toky
- Pomocné toky (len na zahrnutie – include)

Názov prípadu použitia

- Dôležité je, aby prípad použitia mal výstižný názov
- Má vyjadrovať funkcionality – slovesná fráza
- Má znieť akčne: *Vystav faktúru*
- V angličtine je to bežné (členy sa vynechávajú): *Issue Invoice*
- Alternatívou je infinitív: *Vystaviť faktúru*
- Ale bežné je gerundium: *Vystavenie faktúry*
- Neimperatívny tvar je však nebezpečný
 - Môže viesť k identifikácii falošných prípadov použitia a funkcionality dekompozícii (*Správa výpožičiek*)
 - Ak sa už používa, dobré je aspoň skontrolovať, či sa dá zmeniť na imperatívne
- V UML je prípad použitia klasifikátor (ako aj trieda) – preto sa niekedy v názve vynechávajú medzery: *IssueInvoice*

ID prípadu použitia

- Podľa konvencie v projekte – dôvody použitia sú diskutabilné
- Náhrada dlhého názvu prípadu použitia – ale bez zrejmeho významu
- Použitie ID uľahčuje odkazy na prípady použitia, ale zahmlieva význam odkazu
- Prečo takto neoznačíme aj triedy a iné prvky v modeloch UML?
- Najčastejšie UC + číslo (napr. UC008)
- Nástroje väčšinou nepodporujú ID ako také – ID môže byť jedine súčasťou názvu prípadu použitia, čo je problematické
- ID netreba prečíslovať v prípade zrušenia alebo vloženia ďalších prípadov použitia

Krátky opis

- Krátky opis podstaty prípadu použitia
- Väčšinou jeden odsek – rozpracovanie názvu
- Napísať čím skôr po identifikácii prípadu použitia (súčasťou ktorej je pomenovanie)

Základný tok

- Základný tok udalostí (flow of events) v krokoch – označuje sa aj ako scenár
- Musí byť prehľadný a jasný
- Kroky často bývajú očíslované
- Kroky nepredstavujú úplné, ale čiastočné usporiadanie – poradie niektorých krokov sa nemusí dať určiť
- Niekedy sa používajú kľúčové slová podľa konvencie platnej v projekte – napr. vetvenie (*if*) a slučky (*for* a *while*)
- Snahy o úplnú formalizáciu jazyka pre vyjadrenie prípadov použitia nedávajú dobré výsledky
- Snažme sa radšej o presné vyjadrenie v prirodzenom jazyku
- Akceptujme, že sa tieto veci formálne vyjadria v programe
- Obsahuje základnú funkcionálnu – ďalšia je definovaná v alternatívnych tokoch
- Opis prípadu použitia nie je program

Príklad opisu prípadu použitia

UC Zapis do ročníka

Základný tok: Zapis do ročníka

- 1 Študent zvolí zapis do ročníka.
- 2 Systém vyžiada zadanie ročníka.
- 3 Študent zadá ročník.
- 4 Aktivuje sa prípad použitia **Vyber predmety** (jeho rovnomenný tok).
- 5 Študent potvrdí zapis.
- 6 Systém uloží zapis študenta.
- 7 Kedykoľvek počas prípadu použitia, študent môže vzdať zapis, čím prípad použitia končí.
- 8 Prípad použitia končí.

Prípad použitia sa nemusí ukončovať formálnym prehlásením ako v príklade, ale musí byť uvedené ako začína (ak je samostatný).

Alternatívne toky

- Alternative flows, alternate flows, extensions
- Alternatívne toky vyjadrujú zložitú alternatívnu funkcionality
- Často sa používajú na ošetrenie chybových stavov

Príklad alternatívneho toku

UC Zapís do ročníka

...

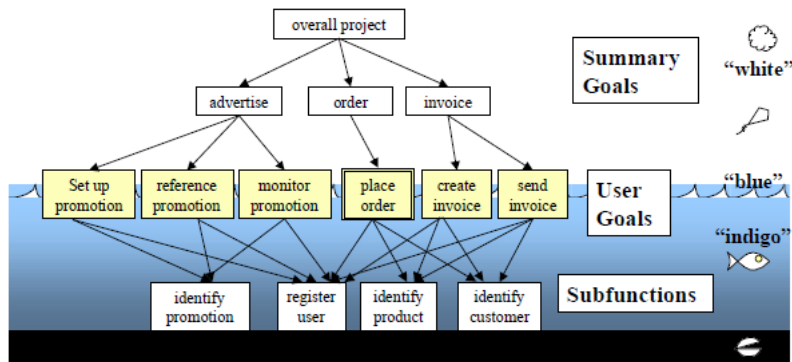
Alternatívny tok: Študent je už zapísaný

Tok sa aktivuje v kroku 1 toku Zapís do ročníka, ak je študent už zapísaný.

- 1 Systém oznámi, že študent je už zapísaný.
- 2 Prípad použitia končí.

V tomto prípade je nevyhnutné uviesť, že prípad použitia končí. Inak sa uvádza ktorým krokom základného toku pokračuje.

Úrovne prípadov použitia



Alistair Cockburn. Writing Effective Use Cases. Addison-Wesley, 2000.

Úrovne prípadov použitia

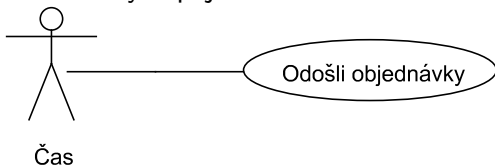
- Prípady použitia, s ktorými pracujeme, opisujú naplnenie konkrétnych používateľských cieľov – to sú pre nás skutočné prípady použitia
- Nad týmito prípadmi použitia možno zostaviť akési sumarizačné prípady použitia s menej konkrétnymi, vyššími cieľmi za účelom získania globálnejšieho pohľadu na systém
- Deje sa to aj naopak: pri tvorbe prípadu použitia zistíme, že je sumarizačný a rozložíme ho do viacerých skutočných prípadov použitia
- Ak zistíme, že sa zaoberáme detailami technickej realizácie, sme pravdepodobne pod úrovňou používateľských cieľov a opisujeme technické funkcie systému

Ďalšie údaje v opise prípadu použitia

- Precondition
 - Predpoklad (označované aj ako vstupná podmienka)
 - Musí platiť pred aktiváciou prípadu použitia; inak nezaručujeme uvedené dôsledky
 - Niekedy silnejšia interpretácia: podmienka aktivácie
- Postcondition
 - Dôsledok (označované aj ako výstupná podmienka)
 - Podmienka, ktorej platnosť zaručujeme, ak pri aktivácii platil predpoklad
- Priorita
- Frekvencia aktivácie

Účastníci v prípadoch použitia

- Prípad použitia aktivuje účastník (niekedy sa hovorí aktér podľa angl. actor)
- Účastník je rola, v ktorej voči systému vystupuje používateľ alebo iný (externý) systém
- Ak sa určitý prípad použitia aktivuje periodicky, ako príslušný účastník vystupuje čas



Účastníci a používateľské práva

- Účastníci nemodelujú používateľské práva
- Medzi účastníkom a prípadom použitia neprebíha žiadna autorizácia a autentifikácia – musia sa vyjadriť explicitne
- Účastník je konceptuálna vec – práva sa dajú konfigurovať
- Účastníci však môžu zodpovedať biznis rolám

Diagram prípadov použitia v UML

Prvky diagramu prípadov použitia

- Prípady použitia
- Účastníci
- Vzťahy medzi nimi
- Predmet (subject)

Vzťahy

- Vzťah medzi účastníkom a prípadom použitia: účastník sa zúčastňuje prípadu použitia
 - Môže to byť aj viac účastníkov – kto prípad použitia aktivuje sa uvádza v opise
- Vzťahy medzi prípadmi použitia
 - Generalizácia
 - «include»
 - «extend»
- Vzťahy medzi účastníkmi
 - Generalizácia
 - Typy účastníkov nie sú vhodné na modelovanie prístupových práv

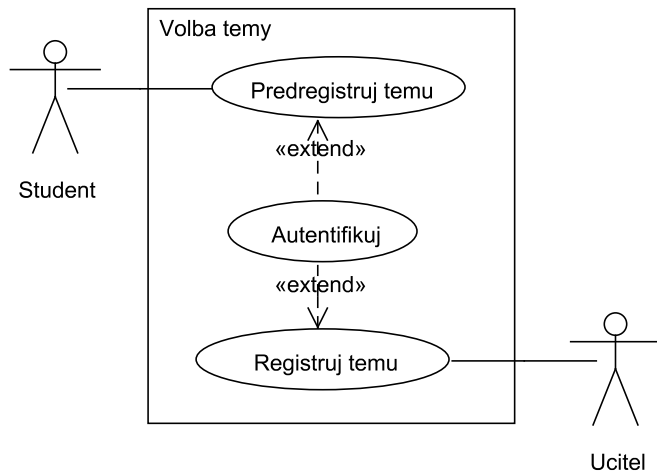
Predmet

- Predmet (subject) je daný systém, na ktorý sa prípady použitia vzťahujú
- Často sa zobrazujú jednotlivé podsystémy («`subsystem`»)
- Často sa vynecháva
- S predmetmi alebo bez nich model prípadov použitia znázorňujeme viacerými diagramami
- Diagram prípadov použitia znázorňuje prípady použitia, ktoré navzájom súvisia
- Ten istý prípad použitia sa môže vyskytovať v rôznych diagramoch

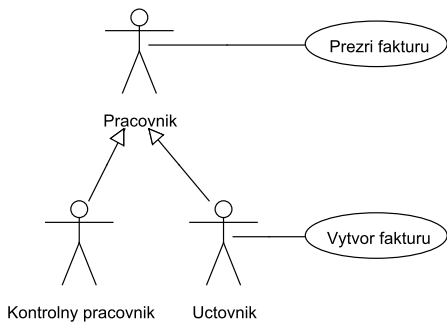
Príklad: Voľba témy bakalárskeho projektu (1)

- Študent si nastaví profil (adresu elektronickej pošty a pod.)
- Študent si predregistruje témy (najviac tri)
- Predregistrované témy študent môže hocikedy zmeniť
- O predregistrovaných témach študent môže konzultovať s učiteľmi, ktorí ich vypísali
- Po dohode učiteľ potvrdí výber témy
- Systém odošle potvrdenie študentovi elektronickou poštou

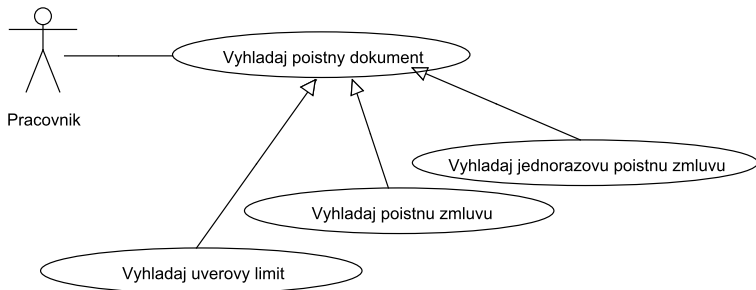
Príklad: Voľba témy bakalárskeho projektu (2)



Príklad: Generalizácia účastníkov



Príklad: Generalizácia prípadov použitia (1)



Príklad: Generalizácia prípadov použitia (2)

UC Vyhľadaj poistný dokument

Základný tok: Vyhľadaj poistný dokument

- 1 Referent zvolí vyhľadanie poistného dokumentu.
- 2 Systém vyžiada kritériá vyhľadávania – úplné údaje alebo ich častí:
 - názov poistného dokumentu
 - meno klienta
 - ...
- 3 Referent zadá kritériá a potvrdí ich zadanie.
- 4 Systém vyhľadá poistné dokumenty, ktoré vyhovujú kritériám a zobrazí ich.
- 5 Referent môže označiť jeden z vyhľadaných poistných dokumentov (implicitne je označený prvý).
- 6 Prípad použitia končí.

Príklad: Generalizácia prípadov použitia (3)

UC Vyhľadaj úverový limit

Základný tok: Vyhľadaj úverový limit

- ① (prekonaný) Referent zvolí vyhľadanie úverového limitu.
- ② (prekonaný) Systém vyžiada kritériá vyhľadávania – úplné údaje alebo ich častí:
 - názov poistného dokumentu
 - meno klienta
 - ...
 - rozsah výšky úverového limitu
 - ...
- ④ (prekonaný) Systém vyhľadá úverové limity, ktoré vyhovujú kritériám a zobrazí ich.

Príklad: Generalizácia prípadov použitia (4)

- Iná možnosť – opisne vyjadriť špecializáciu prípadu použitia
- Dá sa aj naopak: opisne všeobecný (a abstraktný) prípad použitia, a toky v krokoch v špecializovaných prípadoch použitia

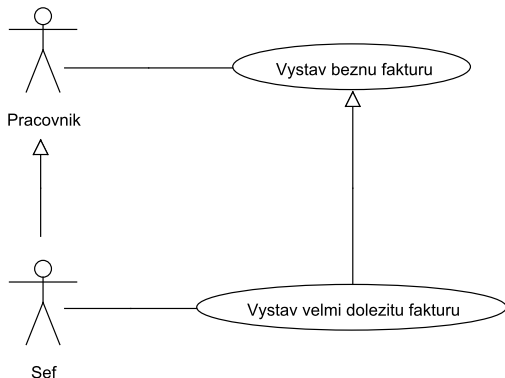
UC Vyhľadaj úverový limit

Základný tok: Vyhľadaj úverový limit

Namiesto všeobecných poistných dokumentov, vyhľadávanie sa týka len úverových limitov. Kritériá vyhľadávania sa rozširujú o údaje špecifické pre úverové limity:

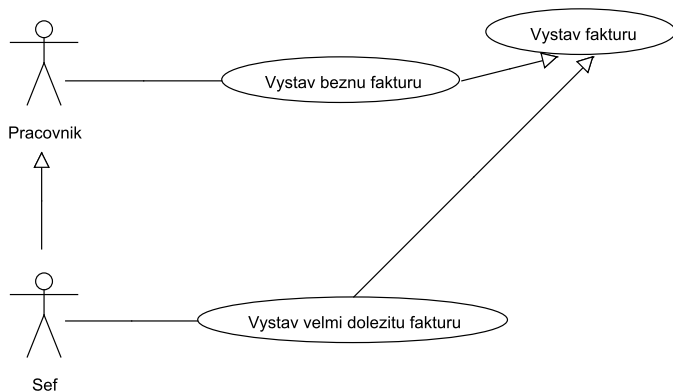
- rozsah výšky úverového limitu
- ...

Príklad: Kombinovaná generalizácia – zlé riešenie!



- Účastník sa zúčastňuje aj všetkých odvodených prípadov použitia

Príklad: Kombinovaná generalizácia – oprava



Použitie vzťahu «extend» (1)

- «extend» – prípad použitia rozširuje iný prípad použitia v bodoch rozšírenia – explicitne vymenovaných alebo deklaratívne špecifikovaných
- Z rozširujúceho prípadu použitia sa nemusia na rozšírenie využiť všetky toky – podmienky ich aktivácie to upresňujú
- Základná funkcionálna rozšíreného prípadu použitia je zvyčajne logicky zachovaná aj bez rozšírenia
- Ale to nemusí znamenať, že je rozšírený prípad použitia zmysluplný v reálnej aplikácii bez rozšírenia

Použitie vzťahu «extend» (2)

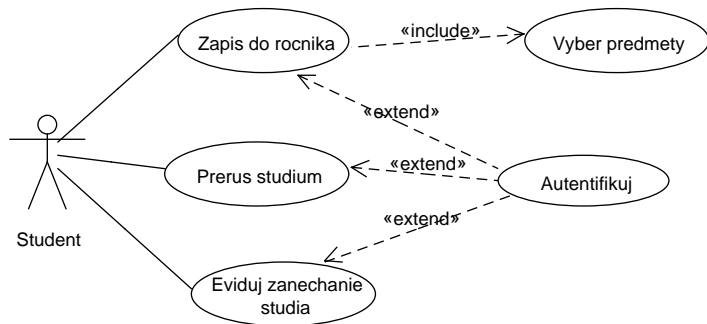
- Najkorektnejšia interpretácia je technická: rozširujúci prípad použitia predstavuje pretínajúcu záležitosť, teda aspekt v zmysle aspektovo-orientovaného vývoja softvéru²
- Rozširujúce toky je preto dobre uvádzať ako alternatívne
- Rozšírený prípad použitia nevie o zásahu rozšírenia – rozširujúci prípad použitia závisí od rozšíreného

²I. Jacobson and P. W. Ng. Aspect-Oriented Software Development with Use Cases. Addison-Wesley, 2004.

Použitie vzťahu «include»

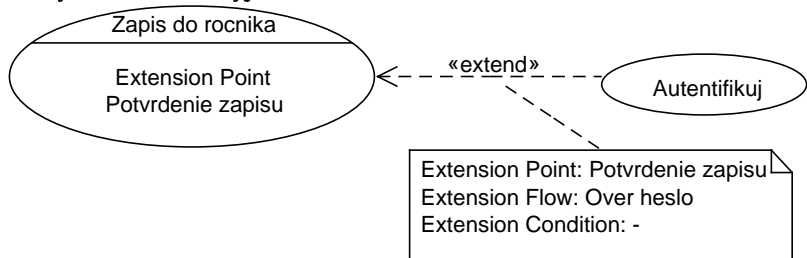
- «include» – prípad použitia zahŕňa iný prípad použitia
- Zo zahrnutého prípadu použitia sa nemusia využiť všetky toky – toky prípadu použitia, ktorý ho zahŕňa, to upresňujú
- Môže to byť funkcionality, ktorú zdieľajú viaceré prípady použitia
- Zahrnutá funkcionality je zvyčajne nevyhnutná aj pre logickú úplnosť prípadu použitia, ktorý ju zahŕňa
- Realizácia krokov zahrnutia v tokoch však môže byť podmienená
- Zahrnuté toky je dobre uvádzať ako pomocné

Príklad: Rozšírenie prípadu použitia (1)



Príklad: Rozšírenie prípadu použitia (2)

- Obvyklé vizuálne vyjadrenie bodov rozšírenia v UML



- Poznámka môže obsahovať aj podmienku rozšírenia (Extension Condition)

Príklad: Body rozšírenia v opise prípadu použitia (1)

UC Zapis do ročníka

Základný tok: Zapis do ročníka

- 1 Študent zvolí zapis do ročníka.
- 2 Systém vyžiada zadanie ročníka.
- 3 Študent zadá ročník.
- 4 Aktivuje sa prípad použitia **Vyber predmety** (jeho rovnomenný tok).
- 5 Študent potvrdí zapis.
- 6 Systém uloží zapis študenta.
- 7 Kedykoľvek počas prípadu použitia, študent môže vzdať zapis, čím prípad použitia končí.
- 8 Prípad použitia končí.

...

Body rozšírenia

- Potvrdenie zápisu: krok 5

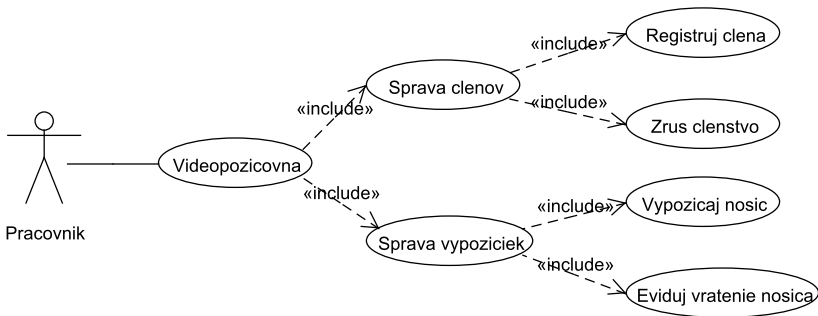
Príklad: Body rozšírenia v opise prípadu použitia (2)

UC Autentifikuj

Alternatívny tok: Over heslo

- 1 Študent uskutočňuje operáciu, ktorá vyžaduje autentifikáciu:
 - **Zapís do ročníka** – Potvrdenie zápisu
 - ...
- 2 Systém vyžiada heslo študenta.
- 3 Študent zadá heslo.
- 4 Ak je heslo správne, systém potvrdí autentifikáciu. Inak ju zamietne a prípad použitia končí.

Ešte ohľadom funkcionálnej dekompozície (1)



Zlý model!

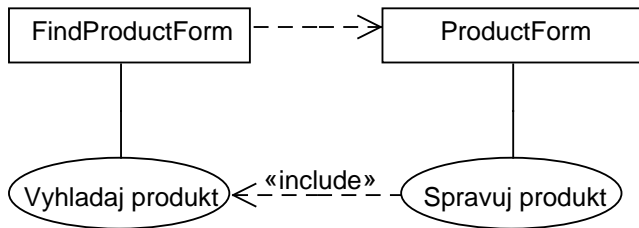
Ešte ohľadom funkcionálnej dekompozície (2)

- Čo je zlé vo funkcionálnej dekompozícii?
- „Vysokoúrovňové“ prípady použitia nepridávajú nič k funkcionalite
- Tieto prípady použitia predstavujú len umelé štruktúrovanie požiadaviek: dá sa robiť viacerými spôsobmi – ktorý je správny?
- Žiaden, lebo OO systém nebude reálne odrážať takúto vhniezdenú reprezentáciu
- Ako vysvetlíme abstraktné, „vysokoúrovňové“ prípady použitia zainteresovaným?

Kedy použiť «include»

- Ak vyčlenený prípad použitia skutočne má povahu prípadu použitia
- Na explicitnú žiadosť zainteresovaných
- Na vyčlenenie spoločnej časti viacerých prípadov použitia alebo ak je to potrebné pre konfigurovanie modelu

Vzťah «include» a GUI



Rozdiely v notáciách

Rôzne notácie modelovania prípadov použitia

- Chápanie podstaty prípadov použitia je pomerne jednoduché
- Notácia modelovania prípadov použitia však nie je jednotná
- Odzrkadľuje aj metodologické rozdiely
- Rozdiely sa týkajú aj grafickej reprezentácie, ale to je druhoradé
- Hlavné sú rozdiely v zápise tokov a spôsobe vyjadrenia krokov

Jacobsonova notácia

- Doterajšie príklady
- Kroky sú očíslované
- Možno mať viac hlavných tokov – vhodné napr. pre CRUD (Create–Read–Update–Delete) prípady použitia (pre každú elemntárnu operáciu samostatný hlavný tok)
- Používanie if vetvenia

Cockburnova notácia

- Kroky sú tiež očíslované ako u Jacobsona
- Vyhýbanie sa *if* vetveniu – najlepšie úplne bez neho
- Dôsledkom je koncízne vyjadrenie základného toku (main success scenario)
- Rozsiahle použitie alternatívnych tokov (extensions) namiesto *if* vetvenia

Övergaardova–Palmkvistovej notácia

- Bez číslovania krokov – opisné vyjadrovanie
- Nadpisy častí prípadov použitia

Príklad Cockburnovej notácie

UC Zapís do ročníka

Hlavný úspešný scenár:

1. Študent zvolí zápis do ročníka.
2. Študent zadá ročník.
3. Aktivuje sa prípad použitia **Vyber predmety** (jeho rovnomenný tok).
4. Študent potvrdí zápis.
5. Prípad použitia končí.

Rozšírenia:

- 1a. Študent je už zapísaný.
- 1a1. Systém oznámi, že študent je už zapísaný.
- 1a2. Prípad použitia končí.

Body rozšírenia v Cockburnovej notácii

- Body rozšírenia v Cockburnovej notácii sa neexponujú
- Rozširujúce prípady použitia sa odvolávajú na miesta rozšírenia opisne v časti označovanej ako Spúšťač (Trigger)

UC Autentifikuj

Spúšťač: Kedykoľvek študent zadá údaje, ktoré sa majú zaznamenať do databázy

Hlavný úspešný scenár:

...

Jacobsonova a Cockburnova notácia (1)

Property	J	C
Single-Step Extension Points	Y	N
Range Extension Points	Y	N
Mandatory Main Flow	N	Y
Multiple Main Flows	Y	N
Subflows	Y	Y
Subflows in Main Flows	Y	Y
Subflows in Alternative Flows	Y	Y
Subflows in Subflows	Y	Y
Alternative Flows	Y	Y
Alternative Flows in Main Flows	Y	Y
Alternative Flows in Subflows	Y	Y
Alternative Flows in Alternative Flows	Y	Y

Jacobsonova a Cockburnova notácia (2)

Property	J	C
Extension	Y	Y
Multiple Extension Locations in an Extension	N	N
Extension by a Specific Flow	Y	Y
Extension Constraint	Y	N
Extension Flow Execution Order	Y	N
Inclusion	Y	Y
Inclusion Constraint	N	N
Inclusion of a Specific Flow	N	N
Description	Y	Y
Typed Steps	N	N
Preconditions	Y	Y
Postconditions	Y	Y
Flow Preconditions	N	N
Flow Postconditions	N	N

Čo ďalej s prípadmi použitia

Vzťahy

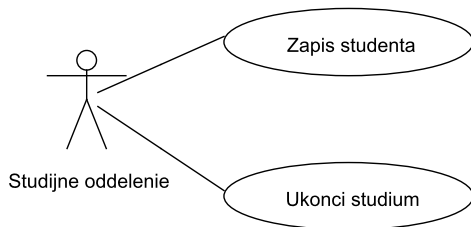
- Prípady použitia definujú správanie systému – možno z nich získať predstavu o jednotlivých metódach, ktoré treba implementovať
- Z prípadov použitia získame aj údajové entity, ktoré ďalej modelujeme diagramom tried alebo entitno-relačným diagramom (ak sa uchovávajú v relačnej databáze)
- Presný opis prípadu použitia diagramom sekvencií
- Formálne sa s prípadom použitia spája prostredníctvom tzv. kolaborácie
- Označuje sa aj ako realizácia prípadu použitia

Diagram sekvencií

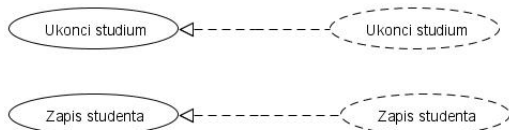
- Diagram sekvencií znázorňuje postupnosť správ prenášaných medzi objektmi
- Príklad – na ďalších slajdoch
 - **:SpravaZapisov** je nepomenovaný objekt triedy **ManazerTem**
 - Objekt **:Student** je inštanciou triedy **Student**, ktorá reprezentuje účastníka v diagrame prípadov použitia

Príklad: zápis a ukončenie štúdia

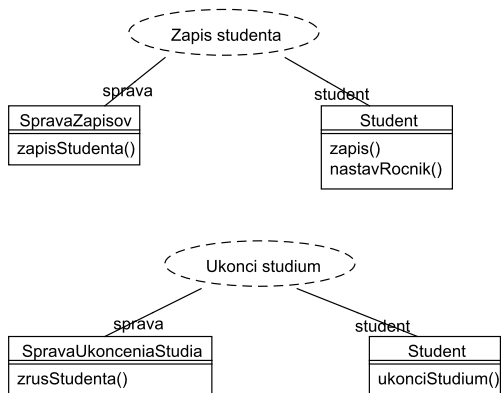
- Prípady použitia:



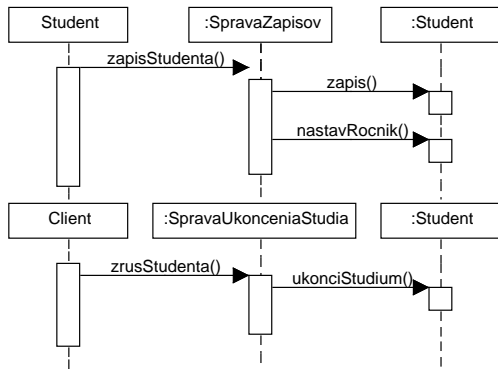
- Zodpovedajúce kolaborácie, ktoré realizujú prípady použitia:



Kolaborácia



Kolaborácia vyjadrená diagramom sekvencií



Sumarizácia

Sumarizácia

- Prípady použitia – opis správania systému z pohľadu používateľa
- Prípady použitia zachytávajú požiadavky – ale nemusia všetky
- Diagramy prípadov použitia poskytujú prehľad
- Primárny je však opis prípadov použitia textom – toky vyjadrené v krokoch
- Bez diagramov sa dá, ale bez opisu nie
- Prípady použitia sa často detailnejšie modelujú diagramami sekvencií
- Prípady použitia slúžia ako podklad pre návrh systému – poskytujú základ pre operácie a identifikáciu údajov
- Do istej miery určujú používateľské rozhranie, ale opis prípadov použitia sa neodvoláva na jeho prvky

Čítanie - základná literatúra

(sčasti alternatívne)

- Jim Arlow and Ila Neustadt. UML and the Unified Process: Practical Object-Oriented Analysis and Design. Addison-Wesley, 2nd edition, 2005. – kapitoly 4 a 5
- Alistair Cockburn. Writing Effective Use Cases. Addison-Wesley, 2000.
- Gunnar Övergaard and Karin Palmkvist. Use Cases: Patterns and Blueprints. Addison-Wesley, 2004. – kapitoly 1–5 a 7–13; dodatočne kapitoly 20, 27, 31, 40, 41 a 44

Čítanie – ďalšia literatúra

- Ivar Jacobson and Pan-Wei Ng. Aspect-Oriented Software Development with Use Cases. Addison-Wesley, 2004. – časť II
- I. Jacobson. Use Cases and Aspects—Working Seamlessly Together. Journal of Object Technology, 2(4):7–28, July-August 2003.
http://www.jot.fm/issues/issue_2003_07/column1
- Ľ. Zelinka. Metamodel modelovania prípadov použitia a jeho profilovanie. Diplomová práca, FIIT STU, Bratislava, 2009. – kapitoly 1–3 a 6 (na dokumentovom serveri AIS; rozdiel medzi Cockburnovou a Jacobsonovou notáciou lepšie pokrytý v článku)
- V. Vranić and Ľ. Zelinka. A Configurable Use Case Modeling Metamodel with Superimposed Variants. Journal of Aerospace Computing Information and Communication (JACIC), AIAA. To appear in a special issue (2010). – na dokumentovom serveri AIS; môžete vynechať časť 5