

Prednáška 3: Od prípadov použitia k štruktúre v jazyku UML

Metódy a prostriedky špecifikácie 2013/14

Valentino Vranič

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

8. október 2013

Obsah prednášky

- 1 Riadenie prípadmi použitia
- 2 Diagramy sekvencií
- 3 Príklad: elektronický obchod
- 4 Diagramy aktivít
- 5 BPMN

Riadenie prípadmi použitia

Unified Process

- Proces softvérového inžinierstva (Software Engineering Process) – metóda
- Plný názov Unified Software Development Process ustúpil pred názvom Unified Process (UP)
- Skorší názov bol Rational Unified Process (RUP) – produkt firmy Rational; určuje štandardy, nástroje atď.; UP ako taký je otvorenejší
- Primárny vplyv mal prístup Ivara Jacobsona – use-case driven
- Vývoj UP a UML bol spojený

Unified Process – charakteristika

- Nebudeme sa zaoberať detailami, ale pripomeňme si hlavné axiómy UP:
 - riadený prípadmi použitia a rizikami
 - sústredený na architektúru
 - iteratívny a inkrementálny
- Prípady použitia zachytávajú správanie
- Riadenie prípadmi použitia predpokladá *odvodenie štruktúry zo správania*

Štruktúra softvérového systému

- Štruktúra softvérového systému: časti, z ktorých sa softvérový systém skladá
- Program je kód, ale aj jeho vykonávanie
- Štruktúra je najpresnejšie vyjadrená kódom
- Ale aj program vo vykonávaní má štruktúru
- Conwayov zákon:

Organizácie, ktoré navrhujú systémy, sú ohraničené tak, aby vyrábali systémy, ktoré sú kópiami komunikačných štruktúr týchto organizácií.

Conway, M.E. "How do Committees Invent?", *Datamation*, (14) 4, April 1968.¹

¹http://egov.blogs.com/eaglossary/2004/06/conways_law.html

Štruktúra má byť odvodená zo správania

- UP: riadenie prípadmi použitia znamená, že sa pri modelovaní najprv sústreďujeme na správanie
- Správanie je zachytené v prípadoch použitia
- Štruktúru možno spoznať priamo v tokoch prípadov použitia
- Vhodným medzikrokom je pokus o vyjadrenie správania technikou, ktorá vynucuje exponovanie štruktúry – diagramy sekvencií
- Diagramy aktivít (ako alternatíva) to neumožňujú v dostatočnej miere – zachytávajú (ak vôbec) len údajové objekty

Aplikácia v semestrálnom projekte

- Správanie je najprv vyjadrené tokmi prípadov použitia
- Toky vyjadríme presnejšie diagramami sekvencií a/alebo aktivít (prvá iterácia)
- Z toho vyplynie základná štruktúra systému (predovšetkým z diagramov sekvencií)
- Nad touto štruktúrou môžeme presnejšie vyjadriť správanie (ďalšia iterácia)
- V ďalších iteráciách prípady použitia môžu byť modifikované alebo môžu byť identifikované ďalšie

Diagramy sekvencií

Hlavné prvky diagramu sekvencií

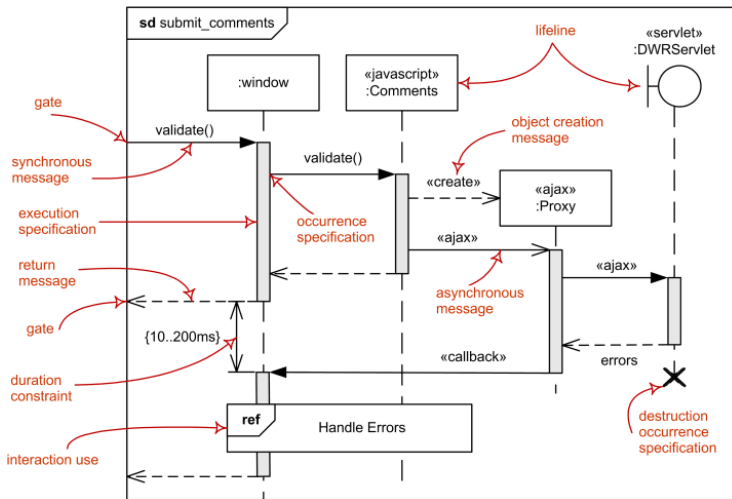


Diagram sekvencií

- Diagram sekvencií znázorňuje postupnosť správ prenášaných medzi líniami života (lifelines)
- Línie života najčastejšie predstavujú objekty
 - Majú názov a typ
 - Môžu byť aj anonymné alebo bez uvedenia typu
- Čas plynie zhora nadol
- Aktívna línia života v danom čase je označená obdĺžnikom
 - Formálne sa označuje ako špecifikácia vykonania – *execution specification*
 - Neformálne sa nazýva aktiváciou alebo blokom aktivácie
 - Nemusí sa uvádzať
- Správa je označovaná názvom operácie, ktorú vyvoláva – s detailami alebo bez nich

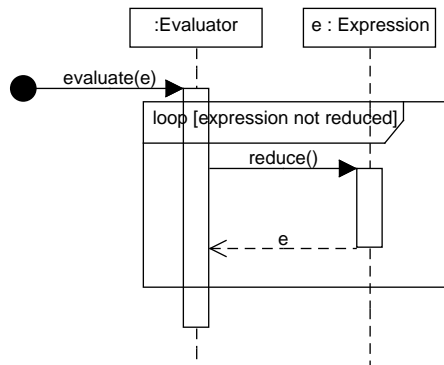
Použitie diagramov sekvencií

- Diagramy sekvencií sa dajú použiť na znázornenie úplného priebehu operácie – často problematické (kód je koncíznejší)
- Skôr majú význam pre čiastkové interakcie – za účelom komunikácie medzi zainteresovanými
- Môžu byť na konceptuálnej úrovni – interakcia objektov konceptuálnych/analytických tried, napr. dá sa vyjadriť:
 - priebeh prípadov použitia
 - interakcia komponentov (reálnych/konceptuálnych) – aj špecifikácia operácií ako takých
 - interakcia biznis entít

Kombinované fragmenty

- Slučky a podmieňovanie sa vyjadrujú kombinovanými fragmentmi (combined fragments)
- InteractionFragment – CombinedFragment
- InteractionOperatorKind:
 - alt
 - opt
 - break
 - par
 - seq
 - strict
 - neg
 - critical
 - ignore/consider
 - assert
 - loop

Špecifikácia operácie (1)

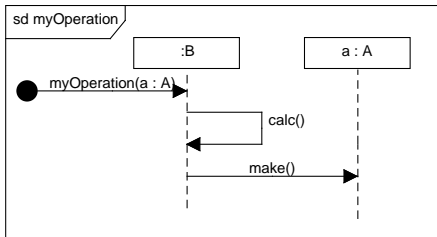


- Ak neberieme do úvahy kontext volania, ale špecifikujeme operáciu ako takú, môžeme jej volanie modelovať nájdenou správou (found message)
- V nástrojoch, ktoré toto nepodporujú (aj IBM RSA pred verziou 8.0), možno použiť nepomenovanú líniu života nešpecifikovaného typu

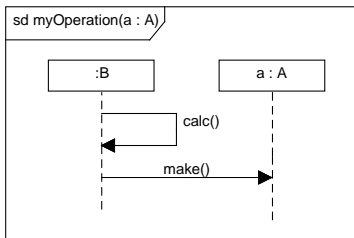
Špecifikácia operácie (2)

- Príklad slučky
- Kombinovaný fragment loop má sémantiku slučky while, teda vykonáva sa kým platí ohraničenie
- Podľa špecifikácie UML ohraničenie má byť mimo ramčeka pre operátor (v ktorom je označenie loop)
- Možno stanoviť aj minimálny a maximálny počet iterácií v zátvorkách pri označení loop: napr. loop(2, 10)

Špecifikácia operácie inak

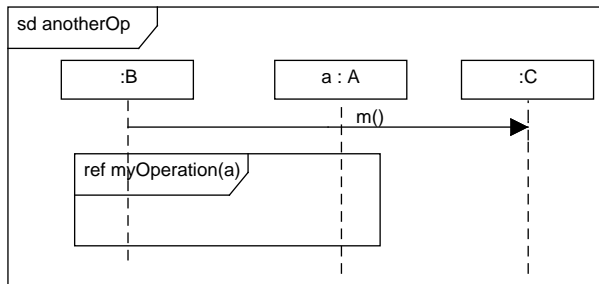


- Diagram ako taký môže mať parametre

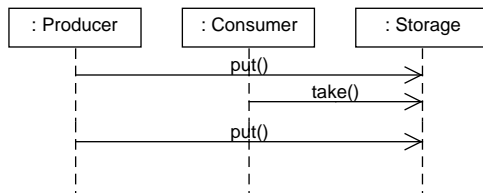


Použitie interakcie v inom diagrame

- Interakciu definovanú diagramom sekvencií možno použiť (vyvolať) v inom diagrame sekvencií
- InteractionUse – ref

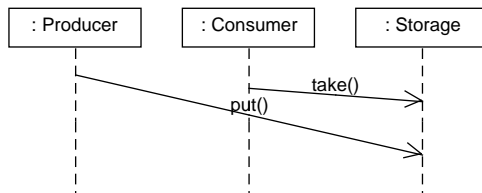


Asynchrónne správy (1)



- Špecifikácia vykonania sa najčastejšie vynecháva

Asynchrónne správy (2)



- Niekedy potrebujeme znázorniť časový posun

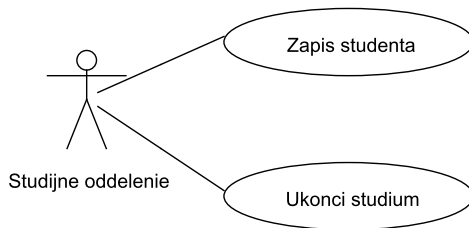
Vyjadrenie priebehu prípadu použitia

- 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, ktorá ho realizuje²

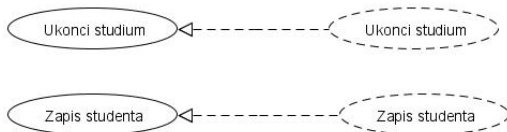
² Kolaborácia sa preto označuje aj ako realizácia prípadu použitia.

Príklad: zápis a ukončenie štúdia

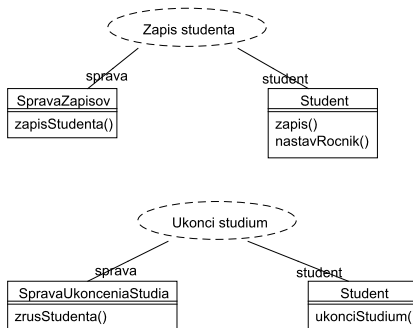
- Prípady použitia:



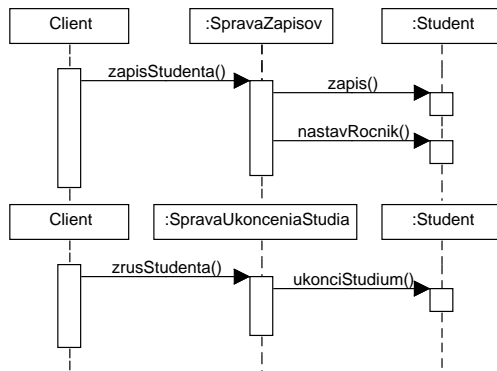
- Zodpovedajúce kolaborácie, ktoré realizujú prípady použitia:



Kolaborácie



Kolaborácia vyjadrená diagramom sekvencií



- Client je objekt bez uvedeného typu
- Dá sa uviesť aj nepomenovaný objekt typu účastníka (: Študijné oddelenie) – presnejšie, ale vyžaduje udržiavanie konzistentnosti s textom

Príklad: elektronický obchod

Príklad: elektronický obchod

- Modelujeme správu produktov v elektronickom obchode

UC Spravuj produkt (1)

Správca spravuje produkt (vytvára, upravuje, zobrazuje alebo ruší – CRUD).

Základný tok: Vytvor produkt

- 1 Správca zvolí vytvorenie produktu.
- 2 Systém vyžiada údaje o produkte.
- 3 Správca zadá názov, typ a obrázok produktu.
- 4 Systém prispôsobí veľkosť obrázku produktu štandardnej veľkosti.
- 5 Správca zaradí produkt do kategórie.
- 6 Ak správca potvrdí zadané údaje, systém ich uloží.
- 7 Prípad použitia končí.

UC Spravuj produkt (2)

Základný tok: Zobraz produkt

- 1 Správca zvolí zobrazenie jestvujúceho produktu.
- 2 Aktivuje sa prípad použitia *Vyhľadaj produkty* (jeho rovnomenný tok).
- 3 Systém zobrazí vybraný produkt.
- 4 Správca prezrie produkt.
- 5 Prípad použitia končí.

UC Spravuj produkt (3)

Základný tok: Uprav produkt

- 1 Správca zvolí úpravu jestvujúceho produktu.
- 2 Aktivuje sa prípad použitia *Vyhľadaj produkty* (jeho rovnomenný tok).
- 3 Systém otvorí vybraný produkt a umožní jeho úpravu.
- 4 Správca upraví údaje o produkte (uvedené v toku Vytvor produkt) vrátane zaradenia do kategórie.
- 5 Ak správca potvrdí zmeny, systém ich uloží.
- 6 Prípad použitia končí.

UC Spravuj produkt (4)

Základný tok: Vyradiť produkt

- 1 Správca zvolí vyradenie produktu.
- 2 Aktivuje sa prípad použitia *Vyhľadaj produkty* (jeho rovnomenný tok).
- 3 Ak správca potvrdí vyradenie produktu, systém na danom produkte nastaví príznak vyradenia.
- 4 Prípad použitia končí.

UC Vyhľadaj produkty

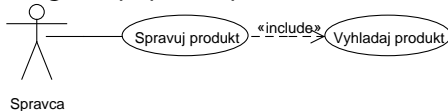
Systém zobrazí zoznam všetkých produktov v kategóriách a umožní označenie jedného produktu.

Pomocný tok: Vyhľadaj produkty

- 1 Systém zobrazí zoznam všetkých kategórií produktov v tvare stromu iniciálne len s prvou úrovňou kategórií.
- 2 Správca môže otvoriť kategóriu a systém zobrazí zoznam produktov zaradených do nej a podstrom jej podkategórií.
- 3 Správca môže zavrieť kategóriu a systém skryje jej obsah.
- 4 Správca môže označiť jeden produkt.
- 5 Ak správca neoznačil žiaden produkt, systém za označený pokladá prvý produkt prvej kategórie.

Prípady použitia a kolaborácie

- Diagram prípadov použitia



- Kolaborácie, ktoré realizujú prípady použitia

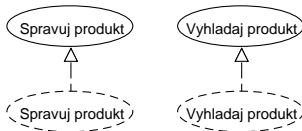
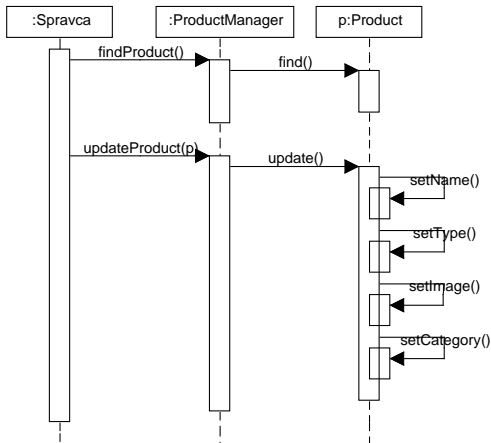
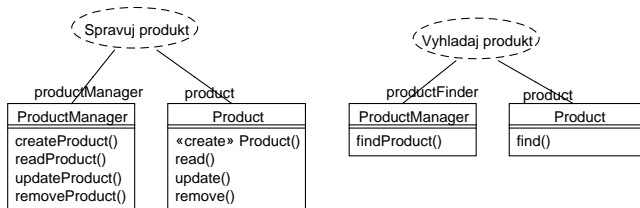


Diagram sekvencií



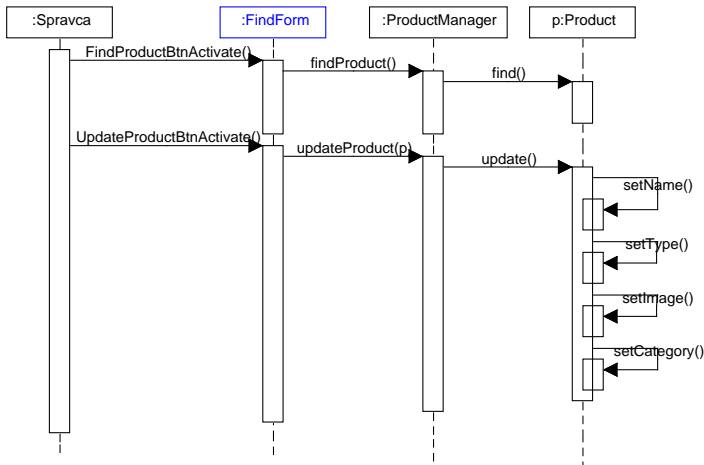
- Tok Uprav produkt

Kolaborácie



- Kolaborácie s vyznačenými rolami, ktorých spoluprácu modelujú
- Roly sú bližšie špecifikované triedami
- V návrhu a implementácii vzniknú triedy, ktoré tieto roly zabezpečujú (aj viac rolí naraz)

Čo s používateľským rozhraním?



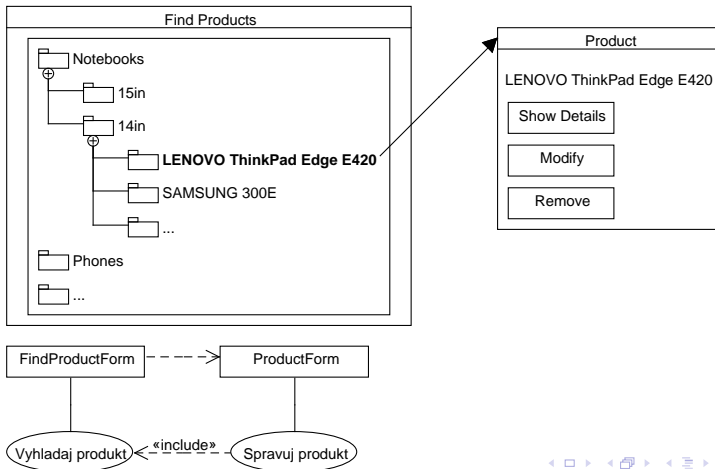
- Približný opis – chýba napr. formulár na úpravu produktu

Prípady použitia a používateľské rozhranie

- Prípad použitia \neq formulár používateľského rozhrania
- Na realizáciu jedného prípadu použitia môže byť potrebná práca s viacerými formulármi používateľského rozhrania
- Viac prípadov použitia môže zdieľať formulár
- Jeden prípad použitia sa niekedy dá realizovať alternatívne prostredníctvom viacerých formulárov

Vzťah «include» a používateľské rozhranie

- Závislosti medzi formulármi používateľského rozhrania nemusia kopírovať závislosti medzi prípadmi použitia



Diagramy aktivít

Diagramy aktivít

- Activity diagrams
- Súčasťou UML
- Pôvodne len lepšie vývojové diagramy, ale od UML 2 majú sémantiku založenú na Petriho sieťach, ktorá určuje pravidlá aktivácie prechodov medzi uzlami
- Používajú sa aj na špecifikáciu tokov udalostí v prípadoch použitia
- Iné použitia:
 - modelovanie biznis procesov
 - znázornenie algoritmov
 - znázornenie vnútornej logiky zložitých operácií

Hlavné prvky

- Diagramy aktivít sú grafy
- Uzly
 - Activities – aktivity
 - Actions – akcie
 - Decision and merge nodes – rozhodovacie a zlučovacie uzly
 - Fork and join nodes – rozvetvovacie a spájacie uzly
 - Objekty
- Hrany sú toky riadenia
- Akcie sú atomické – aktivity majú vnútornú štruktúru
- Akcie môžu predstavovať vyvolanie aktivity, správania alebo operácie, ale aj vyslanie signálu a prijatie udalosti (aj časovej)
- Rozdelenie diagramov aktivít: partitions (niekedy swimlanes) – úseky (drahy)

Prípad použitia diagramom aktivít (1)

UC Zapís do ročníka

Základný tok: Zapís do ročníka

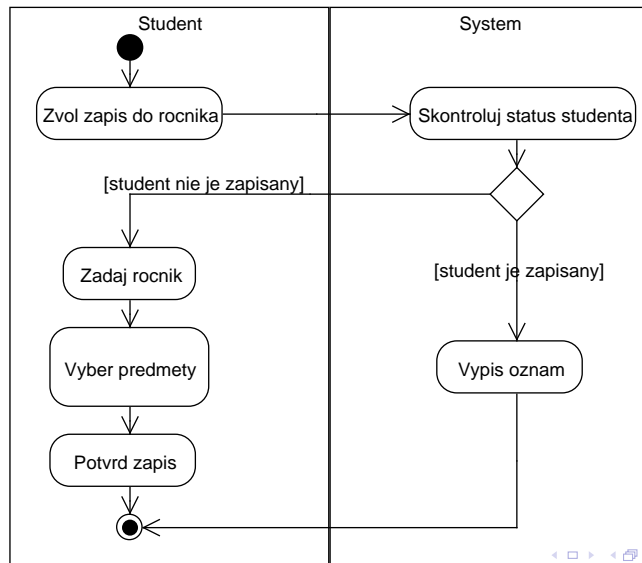
- 1 Študent zvolí zápis do ročníka.
- 2 Študent zadá ročník.
- 3 Aktivuje sa prípad použitia *Vyber predmet* (jeho rovnomenný tok).
- 4 Študent potvrdí zápis.
- 5 Prípad použitia končí.

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.

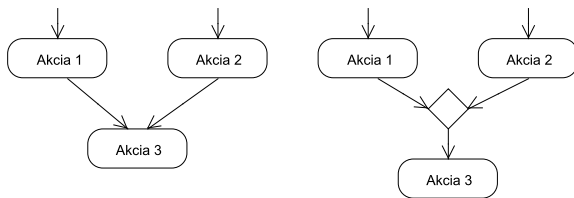
Prípad použitia diagramom aktivít (2)



Aktivácie akcií (1)

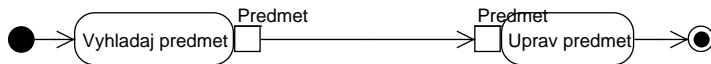
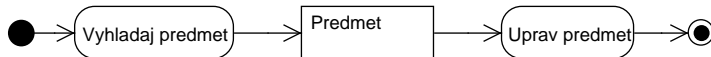
- Diagramy aktivít v UML 2 majú sémantiku Petriho sietí
- Hranami cestujú značky (tokens) a aktivujú akcie
- Počiatočný uzol generuje značku
- Akcia sa aktivuje len ak má značku na všetkých vstupoch
- Po aktivácii akcia vyšle značku na všetky svoje výstupy

Aktivácie akcií (2)

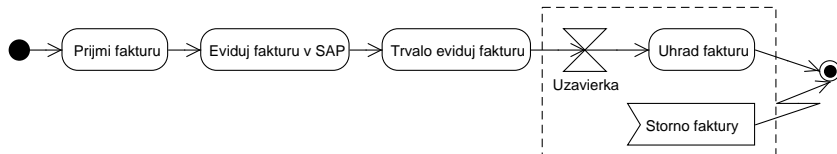


- Bez uzlu spojenia (merge node) sa Akcia 3 aktivuje jedine ak sa uskutočnia aj Akcia 1, aj Akcia 2

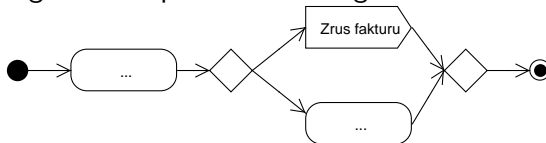
Príklad: objekty



Príklad: prijatie udalosti



- Kedykoľvek počas uzavierky možno zrušiť faktúru
 - Prerušovanou čiarou je vyznačený región prerušiteľnej aktivity (interruptible activity region)
 - Samotná zalomená čiara znamená spracovanie výnimky
- Signál môže prísť z iného diagramu



BPMN

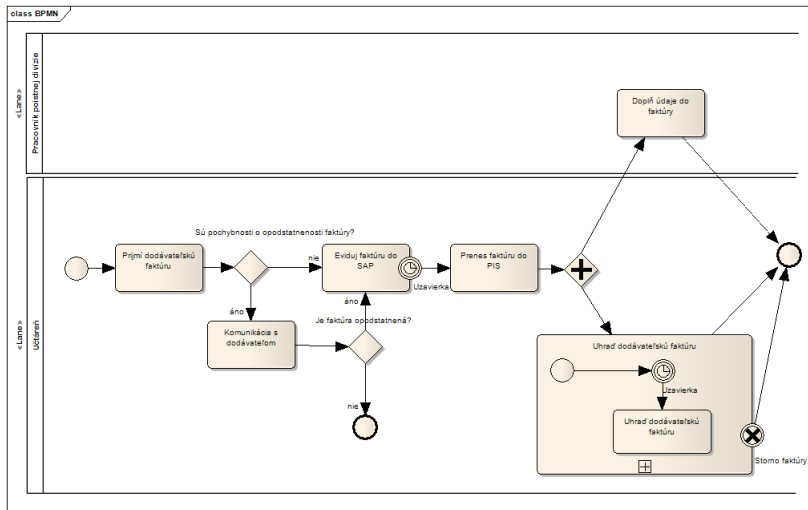
- Business Process Modeling Notation
- Samostatná iniciatíva, ale teraz už pod OMG – OMG Business Process Modeling Initiative, <http://www.bpmi.org/>
- Nie je súčasťou UML, ale má sa stať
- Pôvodne menej formálne chápané, ale v súčasnosti sa blíži k sémantike diagramov aktivít
- Bohatšia notácia – BPMN Poster³

³ http://bpt.hpi.uni-potsdam.de/pub/Public/BPMNCorner/BPMN1_1_Poster_EN.pdf

Hlavné prvky

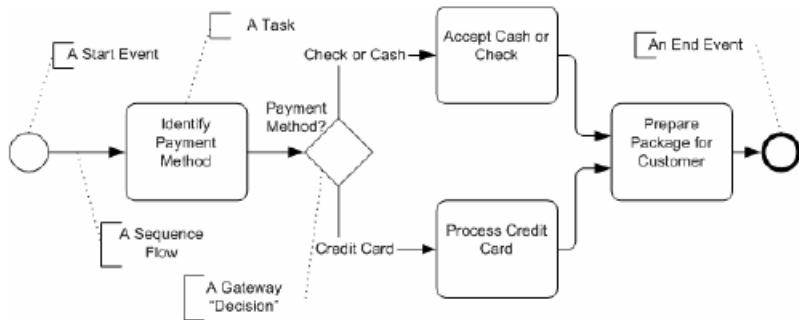
- Activities – aktivity: sub-process and task (podproces a úloha)
- Events – udalosti
- Gateways – brány
- Swimlanes: pools and lanes (!) – dráhy (bazény a dráhy)
- Artifacts

Príklad: prijatie udalosti v BPMN



Príklad

- Riadenie toku v BPMN⁴



⁴ Introduction to BPMN, <http://www.bpmn.org/>

Sumarizácia

Sumarizácia

- Štruktúra má byť odvodená zo správania
- Od prípadov použitia k štruktúre – cez kolaborácie
- Podrobnejšie modelovanie správania:
 - Diagramy sekvencií – vynucuje exponovanie štruktúry
 - Diagramy aktivít – vyjadrujú lepšie kto čo vykonáva
- BPMN – bohatšia notácia na biznis modelovanie, ale náročnejšia na osvojenie všetkých prvkov notácie

Čítanie

- Alistair Cockburn. Writing Effective Use Cases. Addison-Wesley, 2000.
- Kirill Fakhroutdinov. UML Diagrams. <http://www.uml-diagrams.org/>
- Jim Arlow and Ila Neustadt. UML and the Unified Process: Practical Object-Oriented Analysis and Design. Addison-Wesley, 2nd edition, 2005.
- Agile Modeling Home Page, <http://www.agilemodeling.com/>
- Špecifikácia UML, http://www.omg.org/technology/documents/modeling_spec_catalog.htm#UML → Superstructure specification
- BPMN: OMG /Business Process Management Initiative, <http://www.bpmn.org/>