SOFTWAROVÉ INŽENÝRSTVÍ

Požadavky a jejich modelování

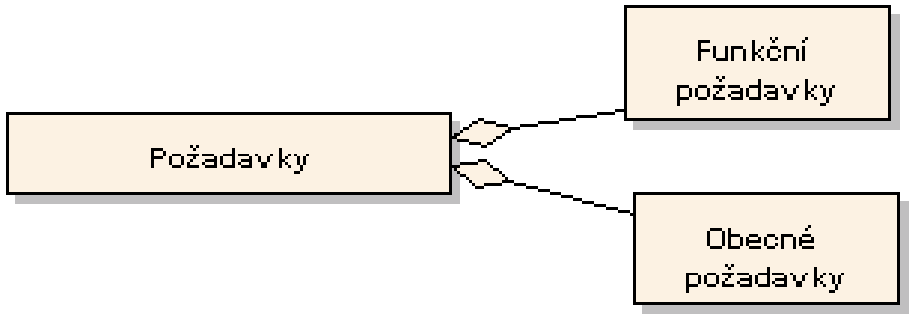
Martin Komárek

Nástroje na týmovou práci

- Revize v textových editorech.
- Administrace Trac.
- EA – požadavky.
- EA – směr čtení názvu asociací.
- EA – mazání elementů z diagramu a modelu.
- EA – generování SQL scriptu.

Požadavky a jejich specifikace

- Zachycují očekávání zákazníka
- mají jasné identifikátory (číslují se)
- hierarchicky děleny (zde na "funkční" a "obecné", ale možno i jinak)



- jeden požadavek=pouze jedno měřitelné "očekávání„
- obsahují (pokud možno) minimální množství informací o implementaci

Funkční požadavky

- definují co bude systém umožňovat
- měly by mít stanoveny priority

Př.:

1.1 Systém bude evidovat(minimálně po dobu pěti let) kdo a kdy měnil emailovou adresu na kterou je uživatelům zasíláno heslo.

1.2. Systém bude vždy při změně emailové adresy, na kterou je uživatelům zasíláno heslo, o této změně uživatele informovat a to tak, že mu na původní i novou emailovou adresu pošle zprávu o tom, kdo a kdy změnil jeho adresu na zasílání hesla.

Obecné požadavky

- vztahují se k celému systému
- spíše omezují způsob, jak bude systém navržen
- většinou se realizují vhodnou volbou jádra systému

Typové příklady obecných požadavků

- požadavky na parametry RAMS (relaibility, availability, maintainability, safety/security)
- požadavky na výkon
- požadavky na použitou platformu (operační systém, hardware, ...)
- požadavky na rozhraní HW i SW s uživateli i jinými systémy (vícejazyčnost, WEB/WAP/SMS brány/... ,touchscreen, RFID čtečka, váha připojena přes RS-232, export zobrazovaných tabulek do souboru ve formátu CSV,…)
- požadavky související s právními aspekty (výhradní / nevýhradní / otevřená licence, otevřený kód, standardy atd.)
- požadavky na usability (čitelnost textu z 1 metru na 15” monitoru 1024*768, Blind Friendly Web Standard, …)
- požadavky na použití Open Source / Open Licence modulů (Log4J, …)

Příklad obecného požadavku

6.5. Systém bude spolehlivý.

Takovýto požadavek může být v závěru projektu oběma stranami interpretován velmi rozdílně. Proto je nutné požadavek upřesnit třeba takto:

6.5. Systém bude spolehlivý.

6.5.1. Střední doba do výpadku systému bude maximálně 20 dní.

6.5.2. Střední doba do opravy systému bude maximálně 12 hodin.

Modelování případů užití

- USE CASE MODEL
- Modelování případů užití, je způsob zachycení **funkčních** požadavků
- Model případů užití obsahuje:

  Hranice systému

  Aktéry (=účastníky)

  Případy užití (=use cases)

  Relace

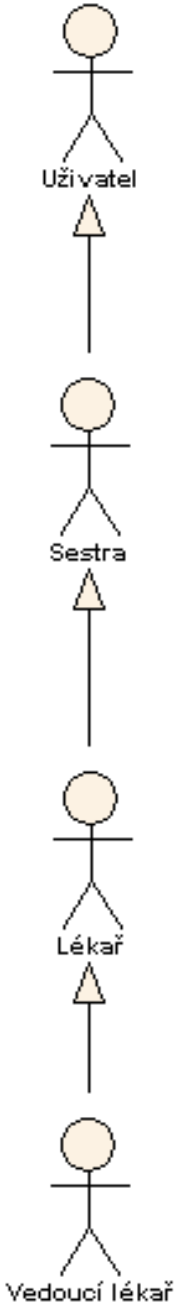
Účastníci (=aktéři)

Jednotlivý účastník(actor)

- jsou vůči systému externími entitou, která systém využívá nebo ho ovlivňuje.

- většinou účastníkem reálná osoba(uživatel), nebo častěji spíše **role**, kterých konkrétní osoby nabývají (př. obchodní zástupce, lékař, hlavní účetní, …..)
- účastníkem ale může být například i "čas" (spouštění záloh atd..) nebo jiný systém

Generalizace účastníků

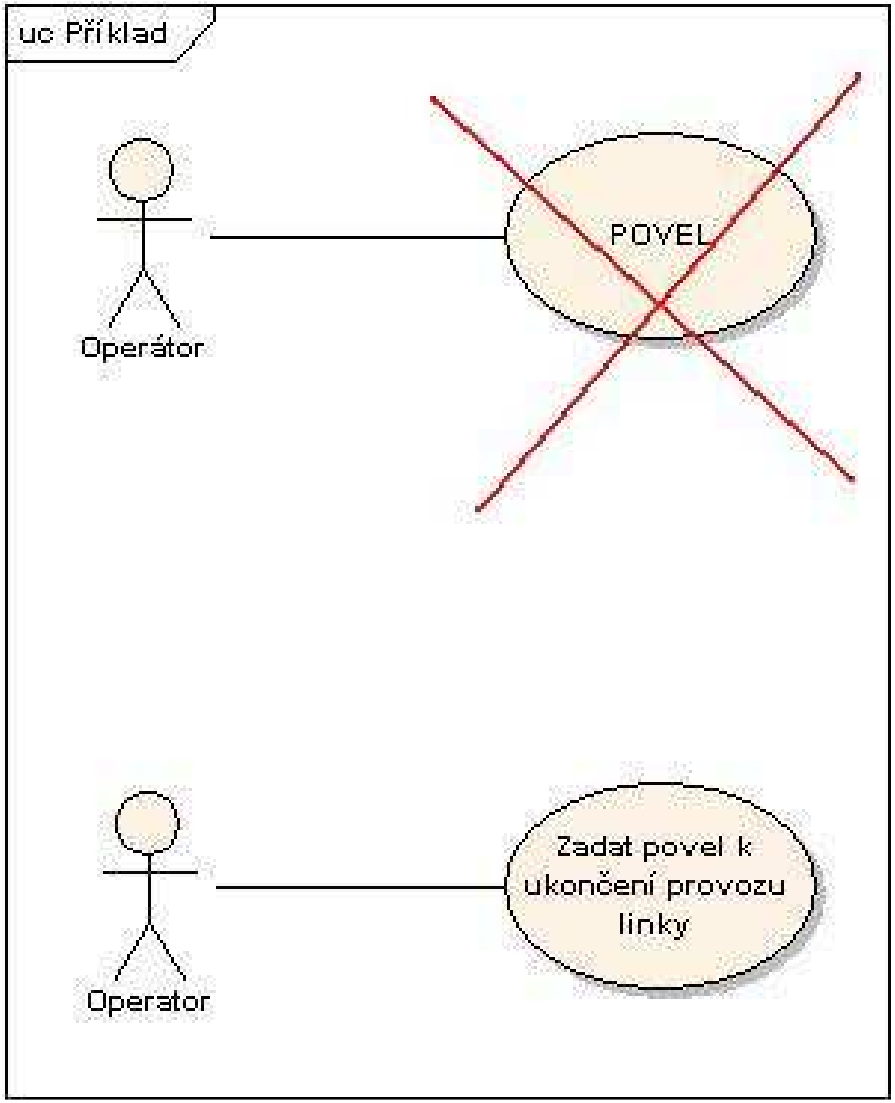


Případy použití systému (USE CASE)

“případ použití"~"případ užití"~"užitná činnost“~”use case”

- James Rumbaugh: „Případ užití můžeme definovat jako specifikaci posloupnosti činností, včetně proměnných posloupností a chybových posloupností, které systém může vykonat prostřednictvím interakce s aktéry.“
- Měl by popisovat jednu rutinní akci jednoho účastníka v jednu chvíli
- Je vždy iniciován účastníkem
- USE CASE diagram znázorňuje funkce systému z pohledu účastníků
- Název USE CASE má vždy *slovesnou vazbu*!!!
- Př. : Zadat objednávku, Zjistit stav objednávky, ……

Název USE CASE !!! Slovesná vazba !!!



Specifikace případu užití

- Struktura specifikace není definována standardem UML.
- Obvykle bývá definován alespoň:

Názvem

Stručným popisem

Hlavním scénářem

Alternativním scénářem (pokud existuje)

Specifikace případu užití

Dále může být definován např.:

- Jedinečným identifikátorem
- Vstupními podmínkami
- Výstupním podmínkami
- Aktéry zapojenými do případu užití
- …
- …

Scénář případu užití = tok událostí

- Popisuje kroky případu užití.
- Odchylky od ideálního (=hlavního) scénáře zachycujeme v alternativních scénářích.
- Většinou textově, ale někdy výhodnější diagramem aktivit (popis algoritmů výpočtu).

Příklad případu užití

Upravit záznam

Umožňuje upravit jednotlivé položky u vybraného záznamu v katalogu. Seznam položek jednotlivých záznamů v katalogu je uveden v popisu tohoto balíčku.

Tok událostí:

Basic Path

Upravení záznamu v katalogu

1. Případ užití začíná, když chce lékař upravit některý ze záznamů v katalogu.
2. INCLUDE (Vybrat katalog)
3. Systém požádá lékaře o výběr záznamu z katalogu, který chce upravovat.
4. INCLUDE (Zobrazit položky katalogu).
5. Systém zobrazí formulář umožňující upravit veškeré položky u vybraného záznamu.
6. Lékař upraví požadované údaje.
7. Systém uloží do záznamu všechny změny provedené lékařem.

Alternate

Zrušení provedených úprav

1. Lékař může veškeré provedené změny odvolat stisknutím tlačítka Storno.

Vztahy mezi případy použití

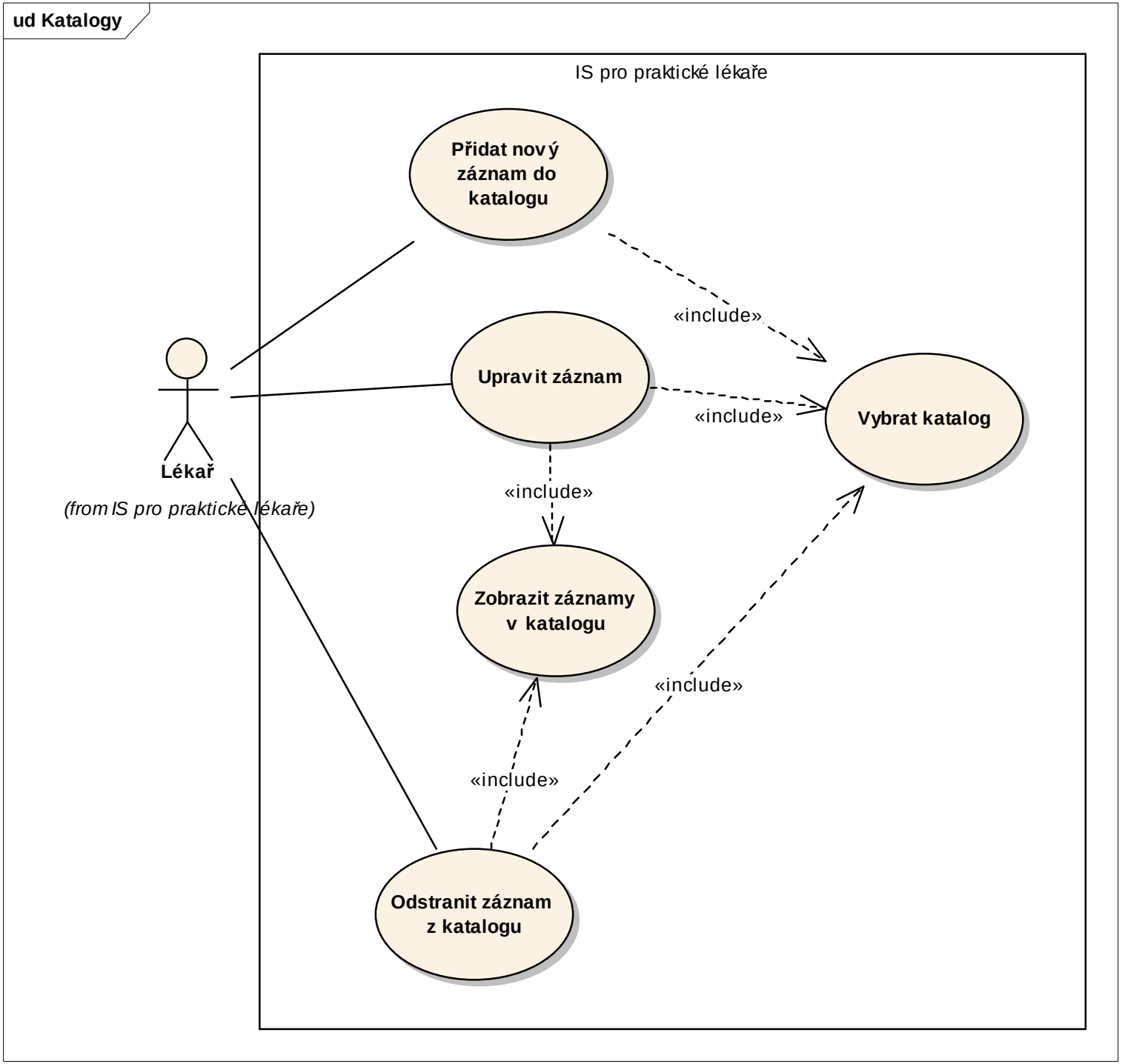
- <<extend>>

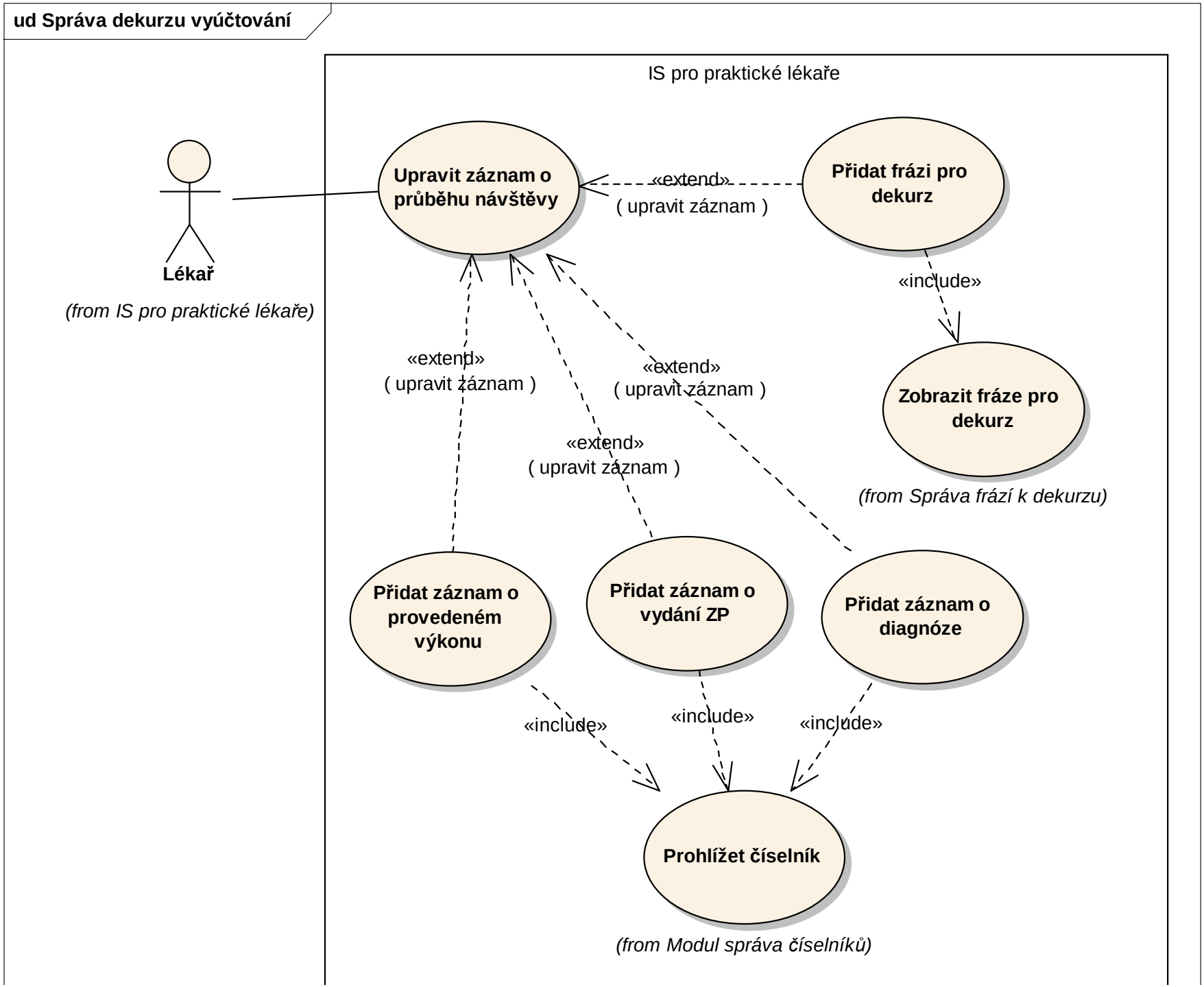
  - pokud nějaký případ rozšiřuje chování (je zde možnost volby)

- <<include>>

  - pokud jeden případ zahrnuje případ jiný

  (např. přepočet ceny prodávaného zboží po změně kurzu EURO)


ud Katalogy
IS pro praktické lékaře
Přidat nový záznam do katalogu
Upravit záznam
Vybrat katalog
Zobrazit záznamy v katalogu
Odstranit záznam z katalogu
«include»
«include»
«include»
«include»
«include»
Lékař
(from IS pro praktické lékaře)


ud Správa dekurzu vyúčtování
IS pro praktické lékaře
Lékař
(from IS pro praktické lékaře)
Upravit záznam o průběhu návštěvy
«extend»
(upravit záznam)
Přidat frázi pro dekurz
«include»
Zobrazit fráze pro dekurz
(from Správa frází k dekurzu)
«extend»
(upravit záznam)
«extend»
(upravit záznam)
«extend»
(upravit záznam)
Přidat záznam o provedeném výkonu
Přidat záznam o vydání ZP
Přidat záznam o diagnóze
«include»
«include»
«include»
Prohlížet číselník
(from Modul správa číselníků)

Matice požadavků a UC

- Sledovatelnost / Tracebility požadavků.
- Ukázka v EA.

7 důvodů proč modelovat UC

Dle Ilji Kravala.
Články 57 až 60 na http://www.objects.cz/clanky/.

1. Velmi vhodné zadání algoritmů chodu aplikace již z analýzy až do programování
2. Výrazné zamezení efektu bobtnání projektu
3. Možné odhady pracnosti na projektu a jeho řízení
4. Efektivnější a snadnější tvorba uživatelské dokumentace
5. Vynikající podklady pro funkcionální testování
6. Podklady pro marketingové materiály a obchodní prezentace
7. Efektivní tvorba dokumentu funkční specifikace produktu jako přílohy smlouvy mezi odběratelem a dodavatelem SW

Odhad pracnosti/rozpočtu pomocí UCP

- UCP = Use Case Points.

Strategické modelování.

- pro potřeby objednavatele.
- pro potřeby rozhodnutí o vstupu do výběrového řízení.

Odhad pracnosti metodou minimaxu

Odhad rozpočtu pomocí COCOMO

- LOC (Lines Of Code)

Package (=balíček)

- Umožňují strukturovat modely.
- Obdoba adresářů.
- Možno použít i pro zachycení softwarové architektury systému.

Dotazy ?

Kdy použít u scénářů „textové dělení“, kdy altrenativní scénaře a kdy diagramy aktivit.

Upravovat zpětně již jednou vytvořené modely?

Bylo rozchozeno forum na dotazy ucitelu.

Odkazy na další články.