Otázka 4 - Analýza - hledání analytických tříd, hledání atributů a stavů, analýza chování a odpovídající diagramy v UML. (A7B36SIN)

Analýza

Pracovní postup Analýza

Analýza v metodice UP zahrnuje architektonickou analýzu, analýzu případu užití, analýzu třídy a analýzu balíčku. Záměrem analýzy je tvorba analytického modelu, který se zaměřuje na to, co systém musí udělat, ale nezabývá se detaily způsobu, jakým to dělá. Analytický model je kolekcí elementů a diagramů modelu UML, je tvořen v jazyku daného odvětví, zachycuje pohled z určitého pohledu, vypráví příběh o požadovaném sytému a je užitečný pro maximální počet uživatelů a zúčastněných osob. Vedle syntaxe tříd, kterou už jsme probrali, máma také syntaxi balíčků (symbol připomínající složku). Považujme balíček za kontejner elementů a diagramů modelu UML. Každý element nebo diagram je vlastním balíčkem. Můžeme jej tedy modelovat jako balíček se stereotypem <<analytickýModel>>, který obsahuje přesně jeden analytický systém (který modelujeme jako samostatný balíček se standardním stereotypem <<analytickýSystém>>) tvořený nejméně jedním analytickým balíčkem. Analytický systém může osahovat mnoho analytických balíčků a každý analytický balíček může navíc obsahovat vnořené analytické balíčky. Výstupem analýzy jsou dva klíčové artefakty:

- Analytické třídy – Tvoří klíčové pojmy v obchodní doméně
- Realizace případů užití – Ukazují, jak mohou instance analytických tříd vzájemně komunikovat za účelem realizovat chování systému udávané případem užití.
- Model analýzy by měl splňovat následující osvědčené postupy:
- Analytický model středně velkého systému obvykle obsahuje přibližně 50 až 100 tříd.
- Do modelu by měly být zahrnuty pouze třídy, které modelují slovníček pojmů probírané domény.
- Nepřijímejme žádná rozhodnutí týkající se implementace.
- Zaměřme se na třídy a asociace – minimalizujme vazby.
- Používejme dědičnost tam, kde existuje přirozená hierarchie abstrakcí.
- Udržujme model co nejjednodušší.

Hledání analytických tříd

Analytické třídy jsou spolu s realizacemi případu užití výstupem aktivity „Analýza případu užití“. Reprezentují abstrakci problémové domény (ze které vzešel první podnět pro vznik požadovaného softwarového systému) a měly by mapovat pojmy skutečného světa (např. *zákazník, produkt, učet*). Problémovou doménou nemůže být jakákoli obchodní aktivita, ale musí vycházet z fyzického hardwaru, který pro správný chod požaduje příslušný software. Uvědomme si, že všechny třídy analytického modelu jsou analytickými třídami a nejsou to teda třídy vzniklé z úvah uvedených při návrhu a pokoušíme se v nich zachytit podstatu abstrakce (zobecnění) a implementační detaily odkládáme do etapy návrhu. Analytické třídy jsou postupně upřesňovány do jedné nebo více návrhových tříd. Hlavním cílem objektově orientované analýzy je nalezení tříd pro dříve zmiňované objekty.

Analytická třída by měla obsahovat pouze ty nejdůležitější atributy (budou je pravděpodobně mít i návrhové třídy). Operace analytických tříd specifikují klíčové služby, které musí třída poskytovat (v návrhu se z nich stanou skutečné metody).

Minimální podoba analytické třídy se skládá z:

- Názvu (Název třídy je povinný).
- Atributů (názvy atributů jsou nepovinné, ale uvedení typů jednotlivých atributů jsou nepovinné).
- Operací (argumenty operace a návratové typy jsou uvedeny pouze tam, kde je to nutné pro pochopení modelu).
- Typů viditelnosti (obvykle se nespecifikuje).
- Stereotypů (mohou být uvedeny, pokud vylepšují model).
- Označených hodnot (mohou být uvedeny, pokud vylepšují model).

Dobrá analytická třída by měla mít tyto vlastnosti:

- Její název odráží její účel.
- Je to hrubá abstrakce, která modeluje specifický element problémové domény.
- Mapuje jasně identifikovatelnou vlastnost problémové domény.
- Obsahuje malou, ale správně definovanou množinu odpovědností (poskytovaných služeb - operací).
- Je velmi soudržná.
- Obsahuje minimum vazeb na jiné třídy.

Důležité je aby analytické třídy vlastnily soudržnou sadu odpovědností (správné rozdělení odpovědností na jednotlivé třídy), jež jsou v souladu s účelem třídy (vyjádřeným jejím názvem) a entitou skutečného světa, kterou daná třída modeluje. Praktické poznatky jsou:

- Třída by měla mít 3 až 5 odpovědností.
- Ke každé třídě by mělo být přidruženo několik dalších tříd, s nimiž by měla tato třída spolupracovat, aby byl celý systém uživateli užitečný.
- Model by neměl obsahovat mnoho malých tříd s malým počtem odpovědností.
- Pozor na procedurální funkce, které se zdánlivě tváří jako třída.
- Pozor na třídy, které se zdají, že zvládnou všechno (názvy jako „*systém*“ a „*řadič*“). Tyto třídy by se měli rozdělit určitých soudržných podmnožin.
- Vyhýbejme se široce rozvětveným stromům dědičnosti, které jsou ve skutečnosti zbytečné.

Při hledání tříd používáme analýzu podstatných jmen a sloves. Podstatná jména a fráze tvořené v textu podstatnými jmény jsou v podstatě vyjádřením tříd a atributů, zatímco slovesa a slovesné fráze jsou vyjádřením odpovědností nebo operací třídy. Musíme si však dát pozor na výskyt homonym (slovo mající stejnou výslovnost nebo hláskování, ale má jiný význam) a synonym (různá slova s podobným nebo stejným významem) a na to jestli je problémový doména správně pochopena nebo definována. Ošidné je při této analýze nalezení „*skrytých*“ tříd (nejsou explicitně zmiňovány). Budeme-li mít problém se sestavením analytického modelu, který napohled nebude dávat žádný smysl, dáme se do hledání skrytých tříd. Vhodné zdroje informací při analýze podstatných jmen a sloves:

- Specifikace doplňkových požadavků (pokud existuje)
- Případy užití
- Slovníček pojmů daného projektu
- Vše ostatní (architektura, dokumenty o představě apod.)

Po shromáždění dokumentace analyzujeme získané materiály zvýrazněním následujících komponent:

- Podstatných jmen
- Spojení několika podstatných jmen (např. *číslo letu*)
- Sloves
- Slovesných frází (např. *ověřit kreditní kartu*)

Seznam těchto komponent porovnáme s obsahem slovníčku pojmů a snažme se vyhledat jakákoli synonyma a homonyma. Tím vytvoříme „kandidátku“ tříd, atributů a odpovědností a pokusíme se o zkušební přiřazení atributů a odpovědností jednotlivým třídám. Pokud jsme našli rovněž kandidátské atributy, můžeme je k třídám také připojit. Během toho získáme určitou představu o relacích mezi jednotlivými třídami. Pokud k výsledku přidáme určité kandidátské asociace, získáme první pokus o model tříd, který můžeme dále analyzovat.
Dalším způsobem hledání tříd je metoda štítků CRC. Tato metoda by se měla používat společně s analýzou podstatných jmen a sloves, případů užití a požadavků a společně s slovníčkem pojmů a s další dokumentací. Začneme označením některých lístečků. Lístečky jsou rozděleny na tři oddíly – v horním oddílu uvedeme název kandidátní třídy, v levém seznam odpovědností a v pravém seznam spolupracovníků. Spolupracovníci jsou dalšími třídami, které mohou s danou třídou spolupracovat, aby společně realizovaly část funkce připravovaného systému. Klíčem k úspěchu je oddělení shromažďování informací od jejich analýzy. Tuto metodu je teda nejlepší aplikovat jako dvoufázovou aktivitu.
První fáze (Shromažďování informací) se zúčastní objektově orientovaní analytici, zainteresované osoby a odborníci v jednotlivých doménách a měla by probíhat v tomto pořadí:

- Rozpoutá se diskuse mezi všemi účastníky. Všechny nápady jsou označeny jako dobré a zapsány. Dál se o nich nediskutuje.
- Členové týmu pojmenují předměty, které pracují v jejich obchodní doméně. Každý předmět je zapsán na lísteček, který je potom nalepen na tabuli.
- Tým uvede odpovědnosti, které dané předměty mohou mít a ty jsou zapsány na příslušný lísteček.
- Označíme třídy, které by mohli pracovat společně. Přeuspořádáme lístečky na tabuli tak, aby odráželi příslušnou organizaci a spojíme je čarami, nebo uvedeme spolupracovníky do oddílu spolupracovníků.

Druhá fáze (Analýza informací) se zúčastní objektově orientovaní analytici a odborníci v jednotlivých doménách. Lístečky vyjadřující klíčové obchodní pojmy označují třídy. Další, které se logicky jeví jako součást jiného lístečku budou atributy. Pokud lísteček není nijak důležitý nebo má nezajímavé chování, pravděpodobně půjde o atribut jiné třídy. Pokud máme pochybnosti o nějakém lístečku uděláme z něj třídu a pak se k němu můžeme vrátit.
Vzhledem k tomu, že hledáme výlučné zobecnění, které vyjadřuje skutečné pojmy problémové domény, můžeme třídy hledat ve skutečném světě jako fyzické objekty (letadlo, lidé), kancelář a kancelářské pomůcky (faktury, objednávky), rozhraní k vnějšímu světu (klávesnice, obrazovky, periferie) a koncepční entity (*VěrnostníProgram* v podobě klubové karty).
Dále musíme výstupy jednotlivých analýz pro hledání tříd sloučit do jednoho modelu UML v nějakém CASE nástroji. Tím vytvoříme první verzi analytického modelu. Postupujme takto:

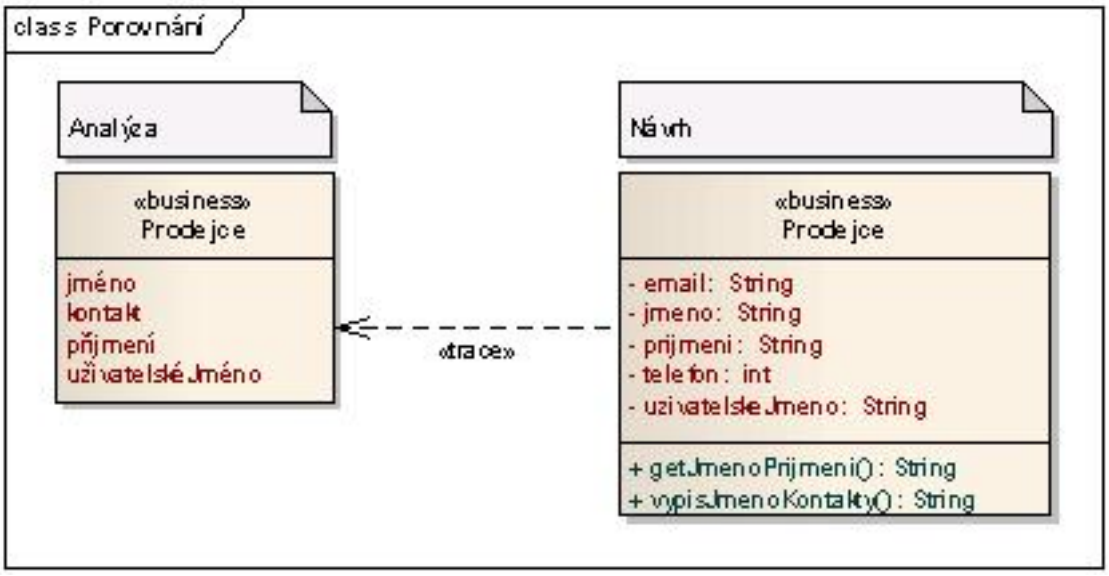
- Porovnejte výsledky analýzy podstatných jmen a sloves s výsledky metody štítků CRC a rozvahy o dalších zdrojích tříd.
- Přeložme synonyma a homonyma pomocí slovníčku pojmů. Najděte rozdíly mezi všemi třemi způsoby nalezení tříd a ihned je vyřešte.
- Vylepšete pojmenování tříd, atributů a odpovědností tak, aby názvy odpovídali standardním konvencím pojmenování firmy.
- Spojte výsledky do první verze analytického modelu.

Analýza chování a odpovídající diagramy v UML

Při tvorbě diagramu tříd je nutné vzít v úvahu jeho účel a rozlišit, zda potřebujeme vyjádřit požadavky na modelovaný software nebo získat podrobný popis designu atd. Z tohoto důvodu se rozeznávají tři úrovně modelu tříd – konceptuální, designová a implementační.

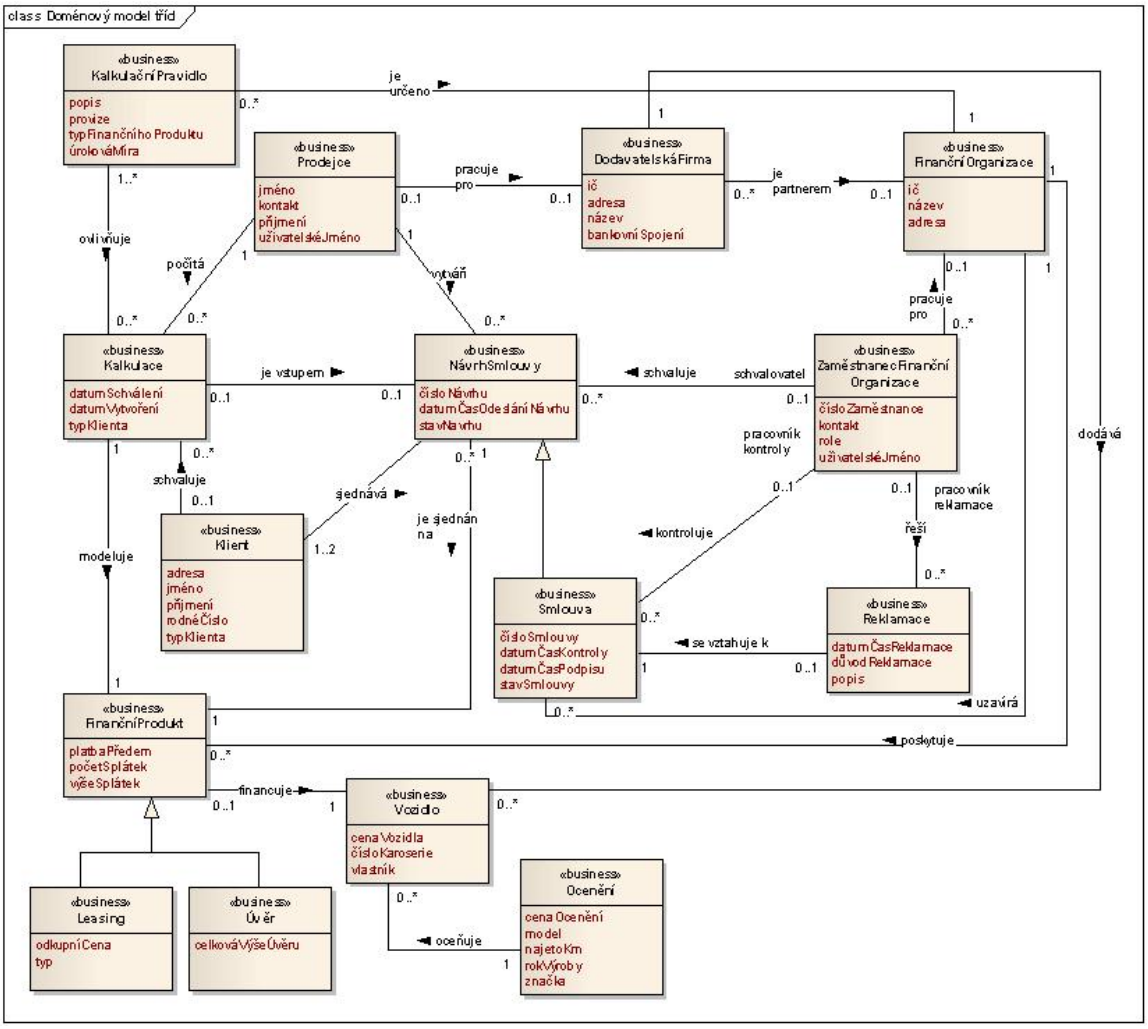
Designový model (model návrhu) vychází z modelu konceptuálního, který rozšiřuje a zpřesňuje například o viditelnosti atributů a metod, datové typy apod. Dále do modelu přidává třídy uživatelského rozhraní (presentation classes) a třídy obsluhující systémové události (control classes). Z jedné třídy v analytickém modelu se tedy může stát v designovém modelu více návrhových tříd. Mezi analytickými třídami a třídami návrhu existuje relace typu trace (viz obrázek 1).

Implementační model se zaměřuje na „grafické zobrazení implementovaného kódu“.



Obrázek 1 - Porovnání třídy analytického a návrhového modelu tříd

Mezi prvky používané v diagramu tříd lze zařadit třídy (classes), asociace (associations), rozhraní (interfaces) a popřípadě balíčky (packages). Třída je „abstrakcí objektů se stejnými vlastnostmi, stejným chováním a stejnými vztahy k ostatním objektům.“ Každá třída má popsány vlastnosti a operace (souhrnně označovány jako features), které může provádět, a omezení definující, jak mohou být jednotlivé třídy propojeny. Vlastnosti tříd jsou v UML označovány jako atributy, operace ve fázi návrhu jako metody (V rámci analýzy se používá označení operace). Třídy jsou vzájemně propojeny pomocí asociací.



Obrázek 2 - Doménový model tříd

Diagram aktivit

Diagram aktivit (Activity Diagram) je typem diagramu interakcí, který se používá pro popis procedurální logiky, byznys procesů (viz obrázek 6) či pracovních postupů. Umožňuje také graficky modelovat jednotlivé případy užití jako posloupnost akcí (viz obrázek 4).

Diagram aktivit prodělal od svého vzniku mnoho úprav a i s novou verzí UML 2.0 došlo k jeho změnám. Zatímco v UML 1.x byl chápan jako speciální případ stavového diagramu, UML 2.0 už tyto dva diagramy nijak nespojuje. Diagram získal novou sémantiku založenou na formálním grafickém modelovacím jazyce Petri Nets (Petriho sítě).

Diagram aktivit modeluje procesy jako aktivity, které se skládají z **uzlů** (nodes) vzájemně propojených **hranami** (edges).

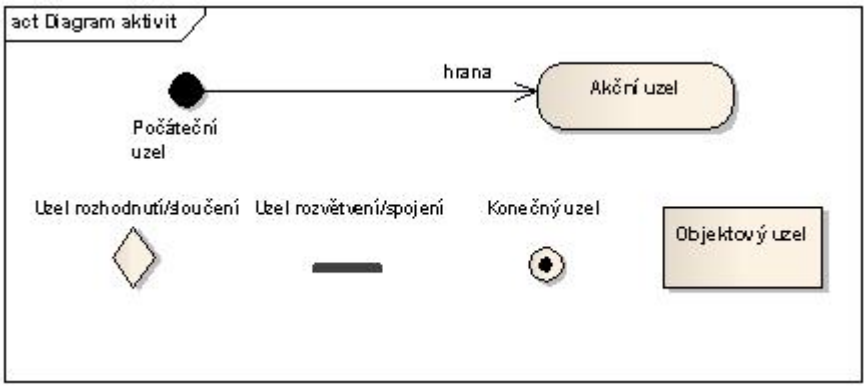
Existují tři typy uzlů – **akční uzly**, které reprezentují samostatné a v rámci aktivity nedělitelné jednotky, **řídící uzly**, jejichž úkolem je řídit cestu uvnitř aktivity a **uzly objektové**, které zastupují objekty. Nejpoužívanějším akčním uzlem je tzv. call action node, který inicializuje aktivitu, chování či operaci. Příkladem řídících uzlů jsou počáteční (initial nodes), konečné uzly (final nodes) nebo uzly rozhodnutí (decision nodes).

Uzel rozhodnutí (v UML 1.x dříve označovaný *branch*) má jednu vstupní hranu a několik konkurujících si hran výstupních. Daný výstup bude zvolen podle toho, která ze vzájemně se vylučujících kontrolních podmínek bude splněna. K označení hrany, která bude použita, pokud nebude splněna žádná kontrolní podmínka, se používá klíčové slovo *jinak* (else).

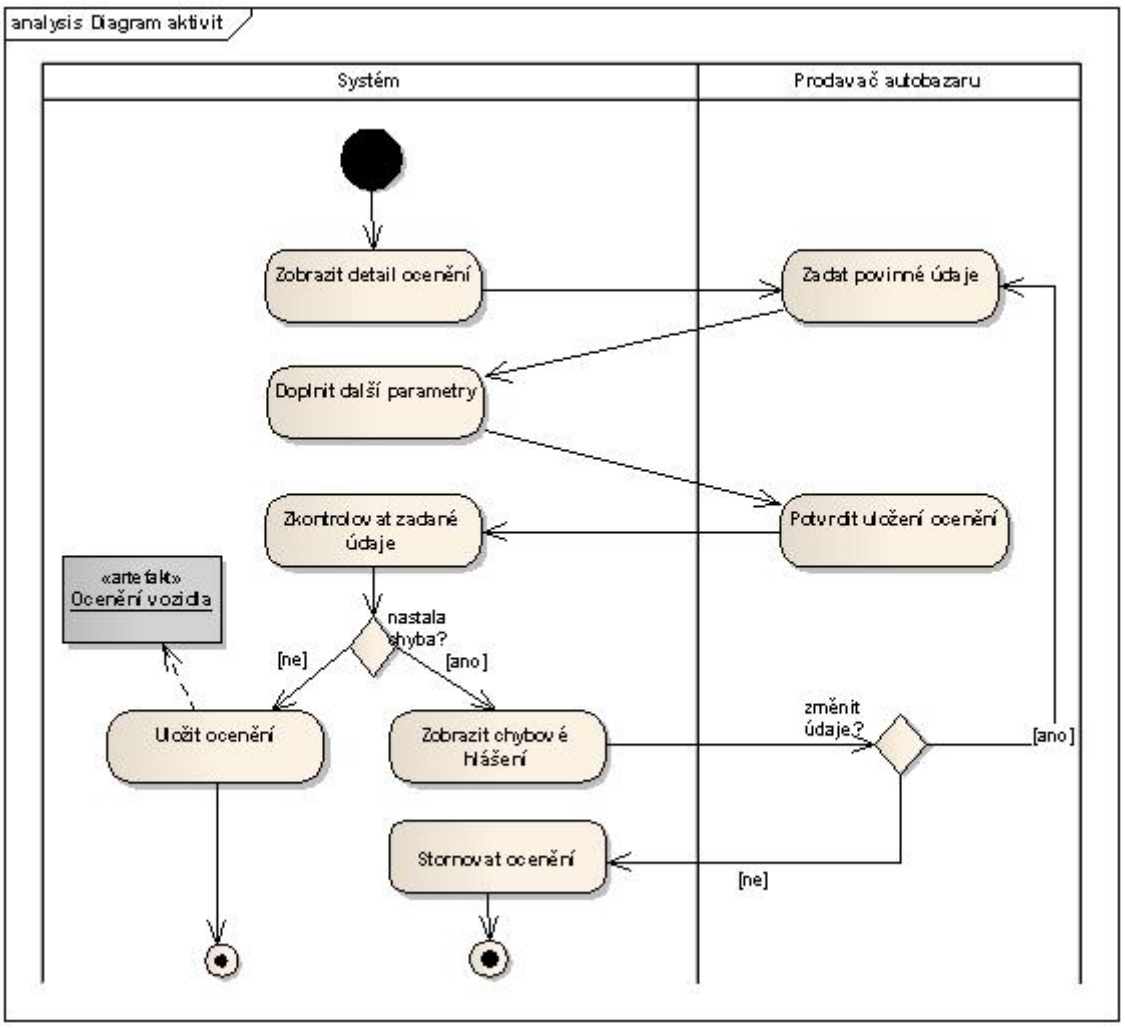
Uzel sloučení (merge) může mít několik vstupních, ale pouze jednu výstupní hranu. Používá se především pro sjednocení větví diagramu aktivit, které byly předtím rozděleny uzlem rozhodnutí.

Procesy, které diagram popisuje, mohou probíhat paralelně, což je umožněno řídícími uzly **rozvětvení** (fork) a **spojení** (join).

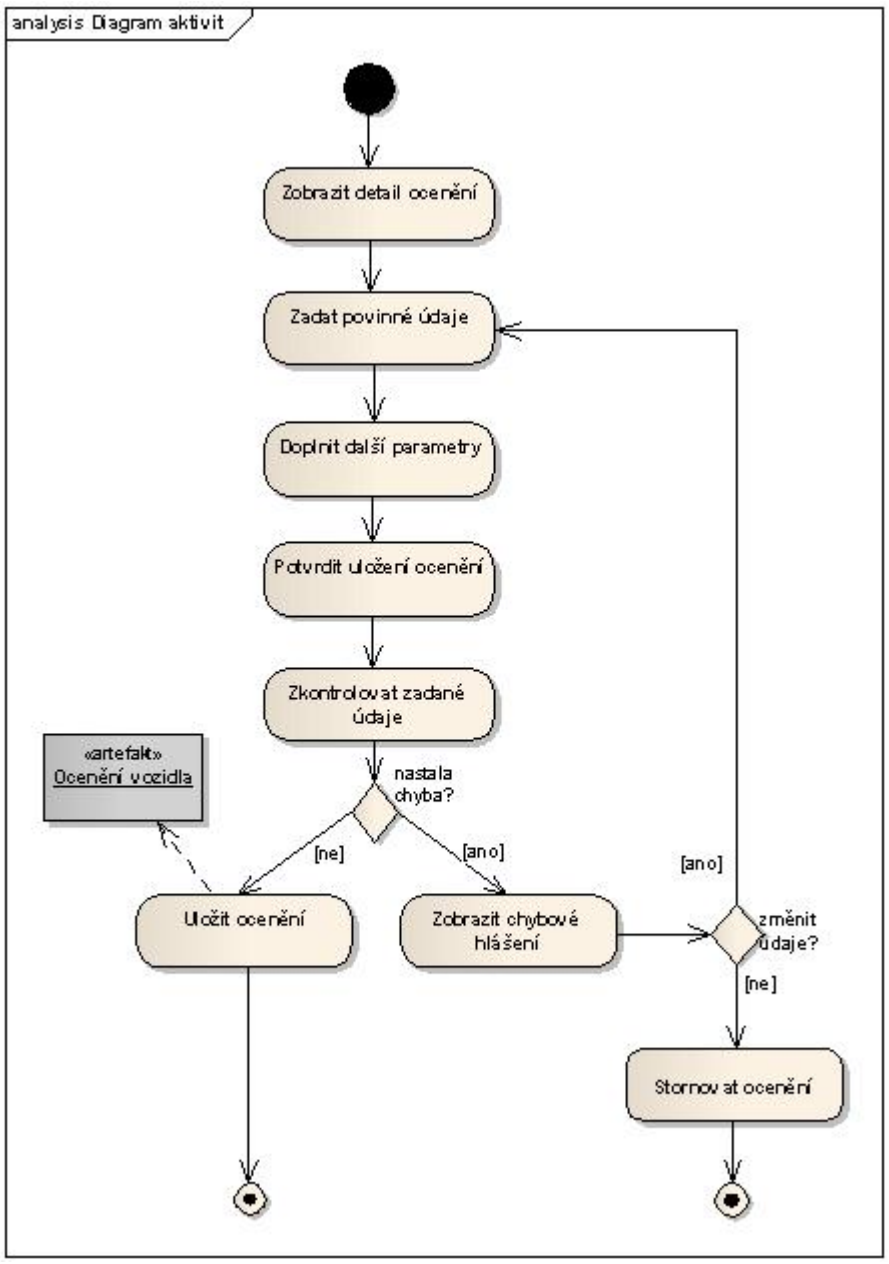
Pro zpřehlednění se může diagram rozdělit například dle rolí (viz obrázek 4) či organizačních jednotek do tzv. **zón odpovědnosti** či **plaveckých drah** (swimlanes). Pro znázornění zón odpovědnosti nabízí UML 2.0 i textovou notaci (viz obrázek 6), která se ale používá pouze v případě, kdy by rozdělení diagramu do zón zhoršilo jeho přehlednost. Obvykle bývá grafický zápis daleko srozumitelnější.



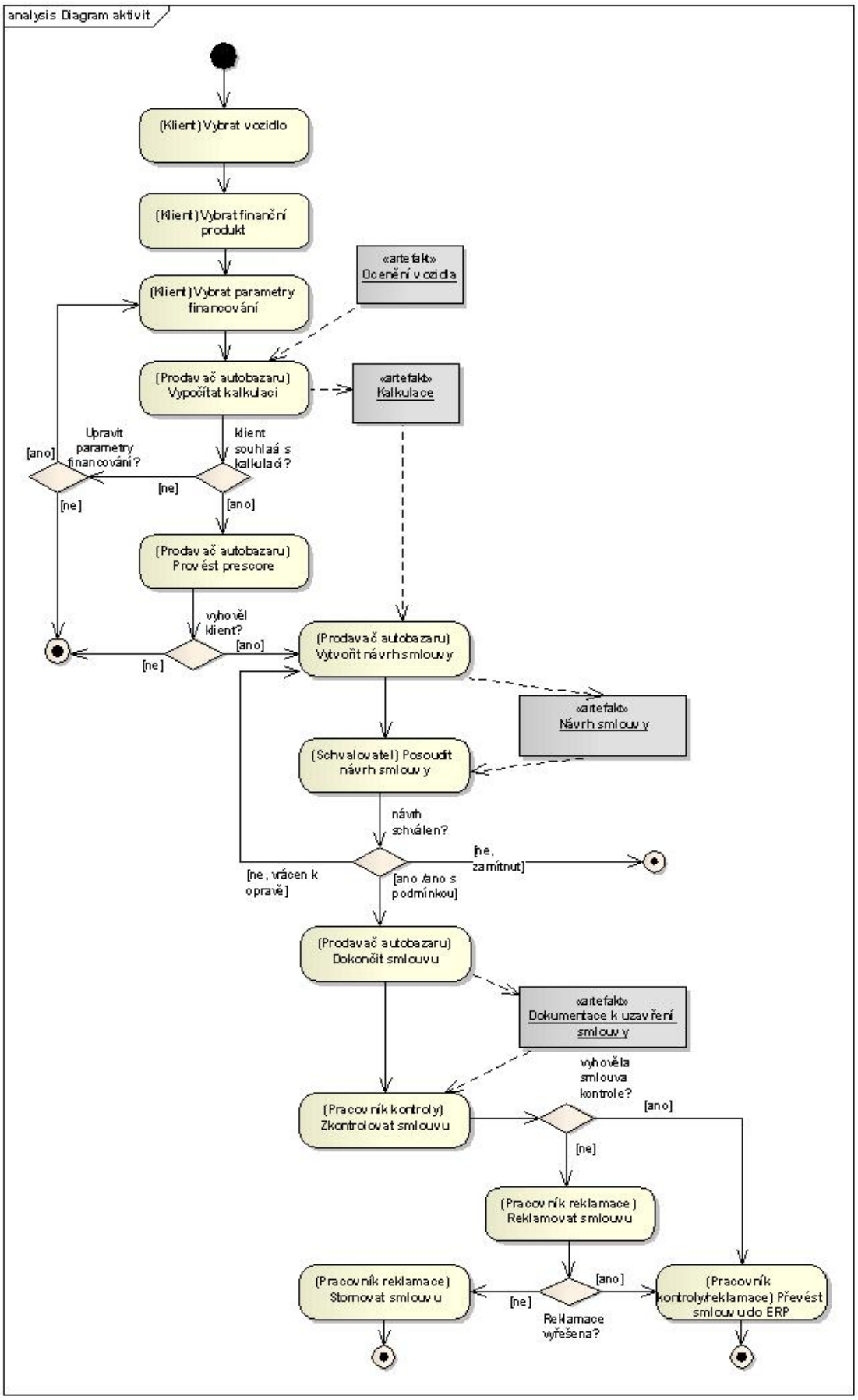
Obrázek 3 - Prvky diagramu aktivit



Obrázek 4 - Použití plaveckých drah v diagramu aktivit (grafický zápis)



Obrázek 5 - Diagram aktivit



Obrázek 6 - Diagram aktivit popisující hlavní byznys proces aplikace

ZDROJE:

http://is.muni.cz/th/72606/fi_b/Bakalarska_Prace_-_Radim_Horak.doc

http://dudka.cz/studyIUS

http://objekty.vse.cz/Objekty/MetodikyANotace-UML

http://plague.cz/statnice/pdf/ais.pdf

http://uml.czweb.org/diagram_trid.htm