

Kapitola 6

Model objektové spolupráce

Po kapitolách věnovaných případům užití a modelům tříd se nyní věnujeme modelům spolupráce objektů nebo chcete-li modelům interakce objektů.

V předchozí kapitole jsme ukázali, jak pomocí analytických tříd můžeme modelovat statickou strukturu systému. Přitom jako jeden z výchozích zdrojů pro identifikaci tříd v systému sloužily popsané případy užití. Nyní využijeme opět případy užití (povšimněte si prosím přínosu správně popsaných případů užití) k tomu, abychom pro ně hledali jejich realizace. Realizací případu užití budeme rozumět takové třídy, které uskutečňují chování případu užití pomocí vzájemné spolupráce. Jedná se vlastně o převod slovního popisu scénáře případu užití na model interakce předem určených tříd.

Je třeba si uvědomit, že proces hledání realizace případů užití je vlastně soustavným (iteračním) upřesňováním. Přitom procházíme specifikace jednotlivých případů užití a modelujeme způsob, jak požadované chování zajistit pomocí nalezené množiny analytických tříd a jimi poskytovaných operací. Volání těchto operací je zajišťováno systémem předávání zpráv mezi objekty.

Pro modelování spolupráce objektů používáme zvláštní typy diagramů, které mají vyjadřovací schopnosti k tomu, aby znázornily, jak mezi sebou objekty spolupracují. Právě interakční diagramy jsou tím nástrojem, který nám pomůže odhalit většinu operací spolupracujících tříd. UML rozlišuje dva základní typy interakčních diagramů:

- sekvenční diagramy (Object Sequence Diagrams).
- diagramy objektové komunikace (Object Communication Diagrams).

Sekvenční diagramy

Vzhled sekvenčních diagramů si ukážeme na případě užití *Založit montážní list* z naší případové studie. Případ užití má scénář podle obr. 6.1.

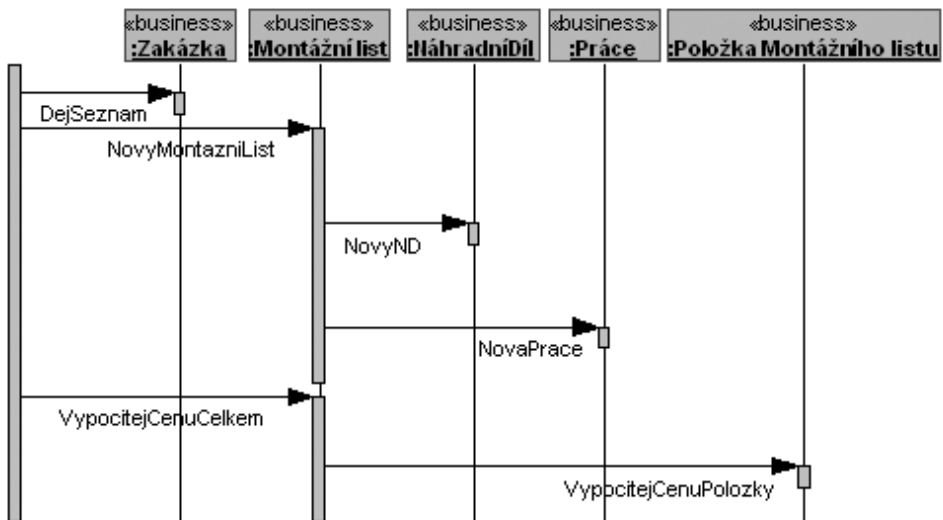
Případ užití: Založit montážní list

Pracovník servisu vystaví montážní list, který obsahuje údaje o opravovaném předmětu: datum odeslání, soupis jednotlivých úkonů (vyměněné součástky, vykázané práce), cenu opravy (DPH), jméno pracovníka, který opravu prováděl, datum opravy. Vytiskne montážní list, který se přibalí k vrácené opravě.

Krok	Role	Akce
1	Uživatel	dá pokyn k zobrazení seznamu existujících zakázek
2	System	zobrazí seznam zakázek
3	Uživatel	vybere zakázku, pro kterou chce vytvořit montážní list, a zvolí funkci <i>Založení montážního listu</i>
4	System	založí nový montážní list pro vybranou zakázku (číslo montážního listu přidělí automaticky) a převezme do něj informace zakázky
5	Uživatel	vloží položky montážního listu, které jsou typu „náhradní díl“ a typu „pracovní úkon“
6	System	uloží zadané položky do zakázkového listu
7	Uživatel	spustí funkci <i>Výpočet ceny</i>
8	System	vypočte cenu použitých náhradních dílů, pracovních úkonů a cenu zakázky celkem
8	Uživatel	spustí funkci <i>Tisk montážního listu</i>
9	System	vytiskne montážní list

Obrázek 6.1 Scénář případu užití *Založit montážní list*

Podle scénáře případu užití na obr. 6.1 byl sestaven sekvenční diagram na obr. 6.2, který si dále popíšeme.



Obrázek 6.2 Sekvenční diagram případu užití *Založit montážní list*

Sekvenční diagramy znázorňují instance objektových tříd (objekty) jako obdélníky na horní hraně diagramu. Na obr. 6.2 vidíte objekty *Zakázka*, *Montážní list*, *Náhradní díl*, *Práce* a *Položka montážního listu*. Z každého objektu vede svislá čára reprezentující život objektu v průběhu chování daného případu užití (bývá také označována jako čára života objektu). Tato forma diagramu byla poprvé popularizována Jacobsonem.

Každá zpráva předávaná mezi objekty je znázorněna šipkou mezi čárami života dvou zúčastněných objektů. Zpráva neznámá nic jiného, než že jeden objekt požádá o vyvolání operace druhého objektu. Např. objekt *Montážní list* odesílá zprávu *NovyND* objektu *Náhradní díl* (což znamená, že objekt *Montážní list* vyvolá operaci *NovyND*, kterou poskytuje objekt *Náhradní díl*). Pořadí, ve kterém jsou zprávy zasílány (a tedy celá sekvence chování musí být v souladu se scénářem případu užití) je dáno směrem shora dolů. Každou zprávu označujeme minimálně jejím názvem, případně předávanými parametry, popř. řídicími informacemi. Poznamenejme, že kromě klasického předávání zpráv mezi dvěma různými objekty lze rovněž znázornit volání objektu sama sebe (tzv. self-call), tedy předávání zprávy, kterou objekt zasílá sám sobě. V tom případě symbol šipky začíná i končí na čáře života stejného objektu.

Pro znázornění aktivity objektu v dané chvíli scénáře můžeme použít aktivační box na jeho čáře života. Vynecháním aktivačních boxů sice můžeme výrazně zjednodušit kreslení diagramů, ale zároveň velmi ztížit jejich chápání. Vzhledem k tomu, že pro modelování interakcí objektů pravděpodobně použijeme některý z nástrojů CASE, nemusíme se otázkou pracnosti kreslení příliš zabývat. Z obr. 6.2 je vidět, že notace sekvenčních diagramů je velmi jednoduchá, a připustíme, že má jistý vizuální půvab.

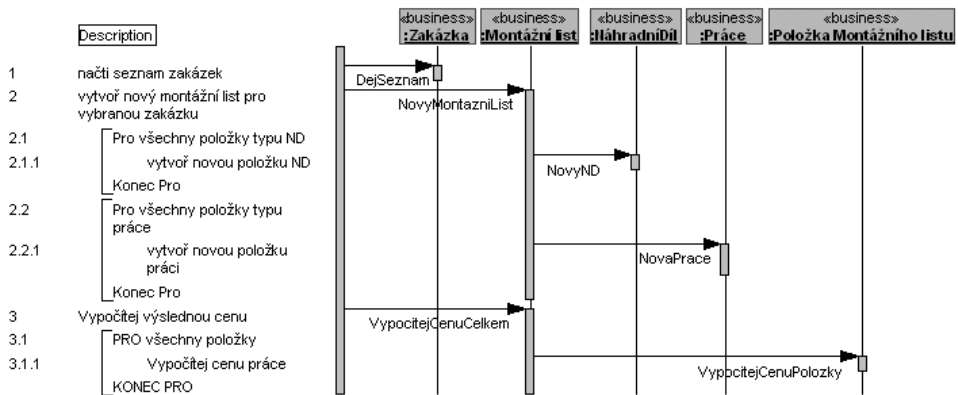
Diagramy mohou rovněž obsahovat znázornění návratů (returns), tedy vrácení aktivity volajícímu objektu, nikoliv zaslání nové zprávy. Vrácení aktivity volajícímu objektu bývá znázorněno přerušovanou čarou. Z praxe ale nedoporučujeme kreslení těchto čar pro jejich malý přínos, vedou naopak ke značnému znepráhlednění diagramů.

Jednou z nejtěžších věcí při porozumění objektově orientovanému programu je pochopení celkového toku řízení. Dobrý návrh respektuje rovnoměrně distribuovanou inteligenci systému v podobě množství relativně jednoduchých metod v rozdílných třídách. Bývá zpravidla složité zachytit celkovou sekvenci chování. Sekvenční diagramy jsou rozhodně nástrojem, který vám pomůže tyto sekvence nalézt.

Dodejme, že sekvenční diagramy můžeme rovněž použít pro zachycení paralelních procesů. Při tom se používá zasílání asynchronní zpráv, pro které je vyhrazen symbol šipky s polovinou hrotu. Po vyslání asynchronní zprávy nečeká volající objekt na dokončení zpracování operace volaného objektu a zpracování pokračuje paralelně. Asynchronní zprávy tak můžete použít na vytvoření nového vlákna (thread), vytvoření nového objektu, popř. komunikaci s vláknem, které již existuje.

Sekvenční diagramy mají také symbol pro zrušení objektu (velké „X“), zpravidla se ale nepoužívá. Důvody jsou podobné jako ve výše uvedeném případě přerušovaných čar: i symboly pro zrušení objektu mají mizivé výhody, zato spolehlivě zhorší přehlednost diagramu. Naopak technikou, která má velký přínos pro celkové pochopení sekvenčního diagramu, je uvedení popisu chování přímo na diagramu.

Metodika Select Perspective, ze které čerpáme a kterou používáme ve své praxi, tuto techniku pokládá za samozřejmou. Popisy chování jsou podporovány také nástrojem CASE Select Component Architect. Je to ukázka velmi užitečného rozšíření standardu UML tam, kde je to potřeba. Jedná se o doplnění sekvenčních diagramů o levý sloupec, obsahující strukturovaný text popisující chování systému. Tento stručný strukturovaný text není částí standardní notace UML, ale ukázal se jako velmi přínosný pro návrháře. Text popisující jednotlivé kroky ve scénáři sekvenčního diagramu může obsahovat běžný popis chování v sekvenci, znázornění podmínek, iterací apod. V rámci tohoto textu, který by měl být stručný a jasný, mohou vývojáři použít prvky jakéhosi metajazyka, např. konstrukty POKUD-JINAK-KONEC_POKUD, PRO-KONEC_PRO apod. Ukázku sekvenčního diagramu z obr. 6.2 doplněného o popis kroků vidíte na obr. 6.3.

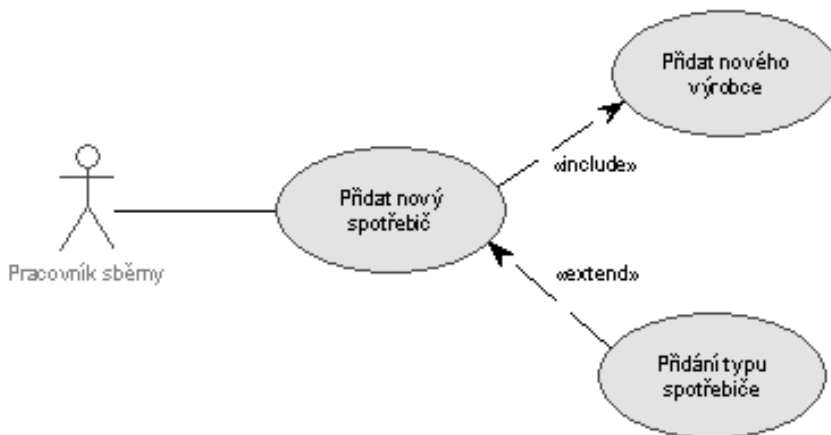


Obrázek 6.3 Sekvenční diagram s popisem kroků scénáře

Sekvenční diagramy umožňují rovněž znázornění vazeb mezi modelovanými případy užití, tedy vazeb typu <<include>> nebo <<extend>>. Na diagramech se k tomu používá symbol sondy. **Sonda** představuje odkaz na jiný případ užití, pro který může být zpracován vlastní sekvenční diagram. Sondy můžeme s výhodou použít tam, kde by sekvenční diagram byl příliš obsáhlý a složitý, což koresponduje s komplexností modelovaného případu užití a jeho rozložením na spolupracující případy užití spojené relacemi. Na obr. 6.4 vidíte diagram případů užití, mezi kterými existují relace <<include>> i <<extend>>. Obr. 6.5 potom znázorňuje sekvenční diagram případu užití *Přidat nový spotřebič* s použitím sond.

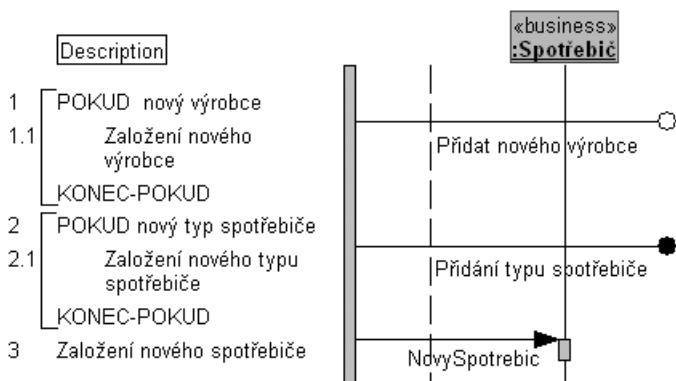
Doplňme, že pro sondu s významem relace <<include>> je vyhrazen symbol s prázdným kolečkem, zatímco pro relaci <<extend>> je kolečko plné, viz obr. 6.5.

Výše popsaná syntaxe modelování podmínek, iterací nebo volání jiných případů užití v sekvenčním diagramu platí pro metodiku Select Perspective. Ve verzi UML 1.4 nebylo možné v sekvenčních diagramech rozumně modelovat podmíněné nebo cyklické chování. Verze UML 2.0 tento problém vyřešila zavedením tzv. rámečků, které ohraničují tu část chování, pro niž platí podmínka, cyklus nebo volání jiného případu užití. Toto volání může být rozpracováno dalším vnořeným sekvenčním diagramem.



Obrázek 6.4 Diagram případů užití s relacemi <<include>> a <<extend>>

Přidat nový spotřebič

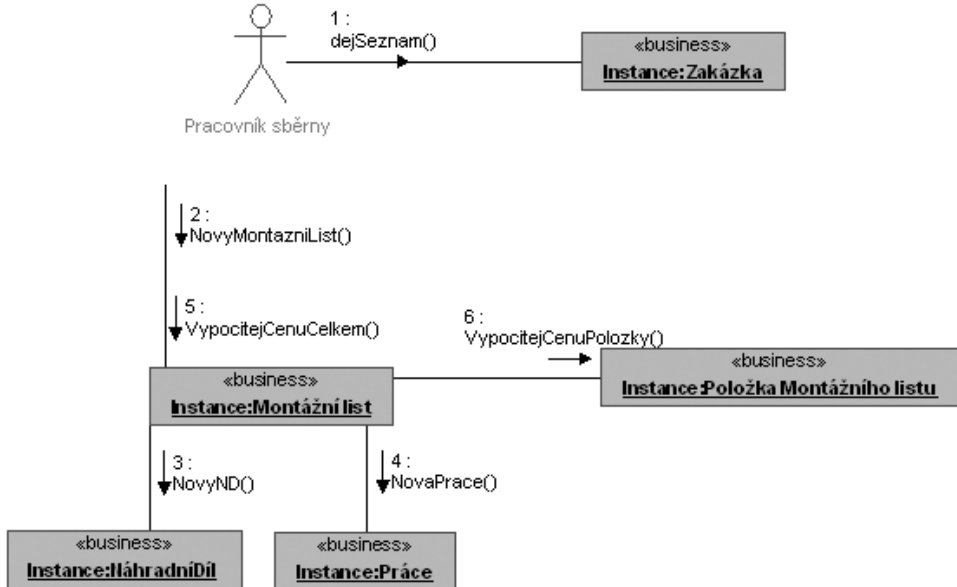


Obrázek 6.5 Sekvenční diagram případu užití *Přidat nový spotřebič*

Diagramy objektové komunikace

Druhou formou interakčním diagramů jsou diagramy objektové komunikace (v literatuře se můžete setkat i starším názvem: diagramy objektové spolupráce). Na těchto diagramech jsou objekty znázorněny obdélníky, vazby mezi nimi pomocí spojnic a šipky indikují zprávy zasílané mezi objekty v rámci případu užití.

Příklad diagramu objektové komunikace, vytvořený pro případ užití *Založit montážní list*, vidíte na obr. 6.6. V podstatě je to jenom jiný způsob grafického vyjádření případu užití. Prvním způsobem je sekvenční diagram.



Obrázek 6.6 Diagram objektové komunikace případu užití *Založit montážní list*

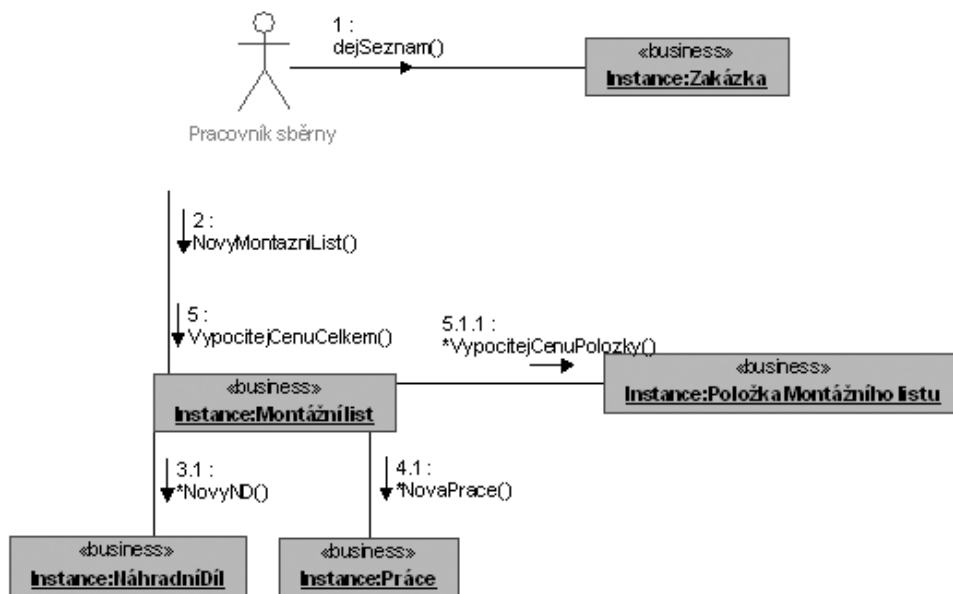
Pořadí zpráv je indikováno jejich číslováním, není tedy tak přehledné a na první pohled patrné jako u sekvenčních diagramů, kde pořadí je dáno sekvencí shora dolů. Na druhé straně forma prostorového vzhledu diagramů objektové komunikace dovoluje lépe znázornit spojení objektů v rámci jejich spolupráce. Vzhled diagramů objektové komunikace může napovědět strukturu balíčků – vyšších celků tříd (packages) na základě komunikace objektů v jednotlivých případech užití.

Pro číslování zpráv na diagramech objektové komunikace lze používat několik schémat. Nejjednodušší je prostá číselná řada, tak jak ji vidíte na obr. 6.6.

Složitější způsob číslování, znázorňující vnoření – předávání aktivity využívá čísel oddělovaných desetinnou tečkou. UML používá právě tento složitější způsob pro lepší možnost znázornění vnoření operací. Daní za tuto výhodu je ovšem složitější orientace v celkové sekvenci volání. Ukázka tohoto typu číslování je na obr. 6.7.

Povšimněte si na obr. 6.7 symbolu hvězdičky (*) pro označení iterace u volání operací *NovyND*, *NovaPrace* a *VypocitejCenuPolozky*.

Pokud jde o formu pojmenování objektů na diagramech podle UML, používá se formát `objectName : packageName.ClassName`, přičemž název objektu nebo název třídy může být vynechán (zejména proto, aby se zmenšila velikost symbolů objektů a tím i celého diagramu). Pokud ale vynecháte název objektu, musíte stejně ponechat na začátku zápisu dvojtečku, aby bylo jasné, že se jedná o název třídy, a ne objektu. Používáte-li některý z nástrojů CASE, nemusíte se o tyto záležitosti zpravidla starat.



Obrázek 6.7 Diagram objektové komunikace s desetinným číslováním pořadí zpráv

Úroveň diagramů interakce

V tomto odstavci se chceme zmínit o úrovni nebo chcete-li hloubce podrobnosti, do které budou diagramy interakce rozpracovány.

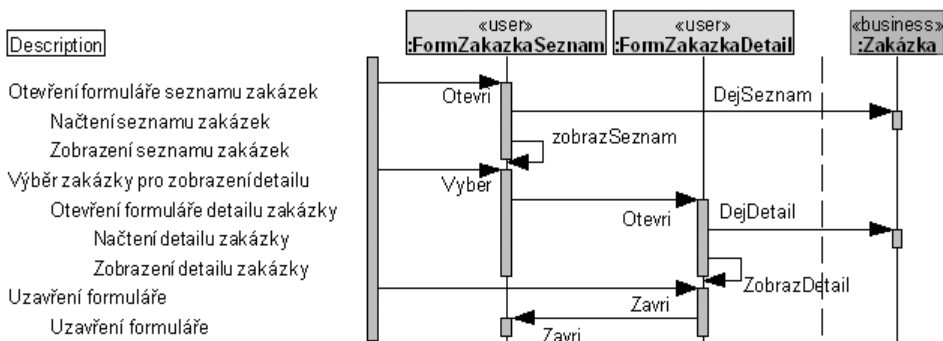
Prvním stupněm je kreslení tzn. analytické úrovně, která se vyznačuje tím, že diagramy obsahují pouze tzv. objekty business. Ukázkou sekvenčního diagramu analytické úrovně jste viděli na obr. 6.2 a 6.3.

Druhým stupněm jsou diagramy, na kterých již je zastoupen návrh (design) v podobě uživatelských objektů – formulářů. Takové diagramy potom znázorňují komunikaci uživatele s formuláři (rozhraním systému) a dále komunikaci mezi formuláři a objekty business. Ukázkou tohoto typu sekvenčního diagramu vidíte na obr. 6.9 a na diagramu objektové spolupráce z obr. 6.10. Pro oba tyto příklady jsme zvolili případ užití *Zobrazit detail zakázky* z důvodu jeho menšího rozsahu. Scénář tohoto případu užití vidíte na obr. 6.8.

Případ užití: *Zobrazit detail zakázky*

Krok	Role	Akce
1	Uživatel	dá pokyn k zobrazení seznamu existujících zakázek
2	System	zobrazí seznam zakázek
3	Uživatel	vybere zakázku, pro kterou chce zobrazit detail
4	System	zobrazí detail vybrané zakázky
5	Uživatel	uzavře formulář detailu i seznamu zakázek

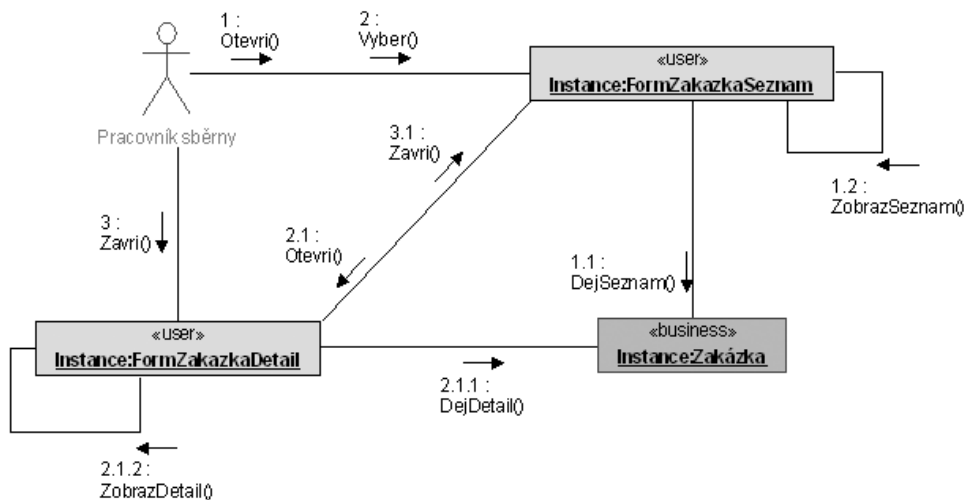
Obrázek 6.8 Scénář případu užití *Zobrazit detail zakázky*



Obrázek 6.9 Sekvenční diagram případu užití *Zobrazit detail zakázky* s objekty rozhraní

Svislá přerušovaná čára mezi objekty rozhraní a objekty business znázorňuje hranici mezi systémem a uživatelským rozhráním.

Povšimněte si korespondujících čísel kroků scénáře mezi sekvenčním diagramem a diagramem objektové komunikace.



Obrázek 6.10 Diagram objektové komunikace případu užití *Zobrazit detail zakázky* s objekty rozhraní

Při rozšíření obou typů diagramů o objekty rozhraní (tzv. users objekty) musíte počítat se zvětšením diagramů a s růstem jejich složitosti.

Třetí úroveň je stav, kdy do diagramů zařadíte i servisní objekty aplikačního frameworku apod. Tím se složitost diagramů dostává k hranici únosnosti.

Naše doporučení k úrovni podrobnosti diagramů je poměrně jednoduché. Podle povahy vašeho projektu, počtu pracovníků týmu a jejich odborné zdatnosti zvolte takovou

úroveň diagramů, aby byl splněn hlavní cíl jejich tvorby: totiž aby programátoři bez problémů rozuměli zadání, co mají naprogramovat.

Porovnání sekvenčních diagramů a diagramů objektové komunikace

Na obr. 6.9 a 6.10 jsou oba zmíněné typy diagramů pro stejný případ užití. Z praktických zkušeností můžeme doporučit spíše sekvenční diagramy, neboť oceňujeme jejich přehlednost v sekvenci zpracování. Je na nich velmi dobře vidět pořadí probíhání akcí a umožňují zapisovat k jednotlivým krokům slovní popis korespondující se scénářem daného případu užití. Není ovšem chybou použít diagramy objektové komunikace, jejich výhodou je možnost zachycení statického propojení objektů a možnost uvažování o balíčcích na základě spolupráce množiny objektů. Principiální výhodou obou typů diagramů je jejich vysoká vypovídací schopnost, nicméně právě s těmito diagramy mívají začátečníci nejvíce potíží.

Doporučení pro tvorbu diagramů interakce

Diagramy interakce můžete použít k modelování chování skupiny objektů v rámci případů užití. Síla těchto diagramů spočívá ve znázornění spolupráce mezi objekty, nejsou však příliš dobré k popisu detailního chování konkrétních objektů. A tak pokud chcete modelovat chování jednoho konkrétního objektu průřezově přes všechny případy užití v systému, je vhodnější použít stavový diagram (State Diagram) – viz kapitola 8. Chcete-li navíc sledovat chování mezi více vlákny, použijte diagram aktivit (Activity Diagram), viz kapitola 9 této knihy.

Pro vlastní kreslení diagramů interakce můžeme poskytnout některá praktická doporučení:

- Máte-li dva podobné scénáře případu užití, lze jejich realizace jistě zachytit na jednom společném diagramu interakce s tím, že použijete větvení a podmínky. Raději ale nakreslete dva samostatné diagramy, budou mnohem jednodušší a přehlednější.
- Problémem diagramů interakce na větších projektech je, že zpravidla není možné je kreslit pro všechny případy užití, a to ani s podporou nástroje CASE. Pracnost s tím spojená zpravidla převyšuje přínosy. Doporučujeme proto nakreslit diagramy interakce pro klíčové případy užití, tj. takové případy užití, které podporují tvorbu základních hodnot v systému, a dále pro takové případy užití, které jsou zástupci určitých skupin případů užití s velmi podobným chováním. Například v systému bude probíhat údržba číselníků čtyř typů (vždy s možností založení nového záznamu, editace a zrušení záznamu). Chování případů užití popisujících správu těchto číselníků budou naprosto stejná a budou se lišit pouze názvy číselníků, potom jistě stačí namodelovat správu pouze jednoho číselníku (jako zástupce, jistý vzorový případ) a ostatní případy užití pouze odkázat na uvedený

vzor. Na tomto místě je ale třeba zdůraznit, že čím více případů užití nebude modelováno pomocí diagramů interakce, tím větší nesoulad mezi modelem případů užití a modelem tříd může nastat.

- V rámci týmu si dohodněte a dodržujte pravidla pro zápis strukturovaného textu, popisu kroků sekvenčních diagramů; jednak jde o slovní spojení (tzn. bude se všude uvádět spojení v rozkazovacím způsobu, např. „otevři ...“, „vypočítej ...“ anebo naopak všechna spojení budou používat názvu činnosti, např. „otevření ...“ a „výpočet ...“) a jednak o formu zápisu větvení (konstrukty POKUD-JINAK-KONEC_POKUD) a iterací (konstrukty PRO-KONEC_PRO).
- Další doporučení se týká velikosti diagramů a úzce souvisí s komplexností případů užití. Nevytvářejte příliš velké a rozsáhlé diagramy interakce, klesá tím jejich srozumitelnost a jejich kreslení i pomocí nástrojů CASE se stává problematické. Raději zvolte vyšší úroveň atomizace případů užití, čímž zároveň klesne velikost diagramů interakce. S výhodou používejte sondy – odkazy na jiné případy užití.
- Domluvte a dodržujte v rámci týmu úroveň diagramů interakce, sledujte jediný cíl: diagramy musí být srozumitelným zadáním pro programátory.

Přínosy CASE nástrojů pro tvorbu diagramů interakce

Je samozřejmé, že nástroje CASE vám výrazně usnadní kreslení diagramů a přinesou některé další výhody spojené s uložením všech informací ve společné repository. Pro ilustraci se zmíníme o několika výhodách nástroje Select Component Architect, který sami používáme.

Samozřejmou výhodou je mnohem snazší kreslení diagramů a jednoduché překreslování bez jakýchkoliv omezení. Dalším přínosem je provázání jednotlivých modelů a jejich prvků, např. případ užití – diagram interakce apod. Samozřejmostí je promítání informací z diagramů interakce (především operací) zpět do diagramu tříd. Ukázka pracovního prostředí pro modelování sekvenčních diagramů je na obr. 6.11.



Shrnutí:

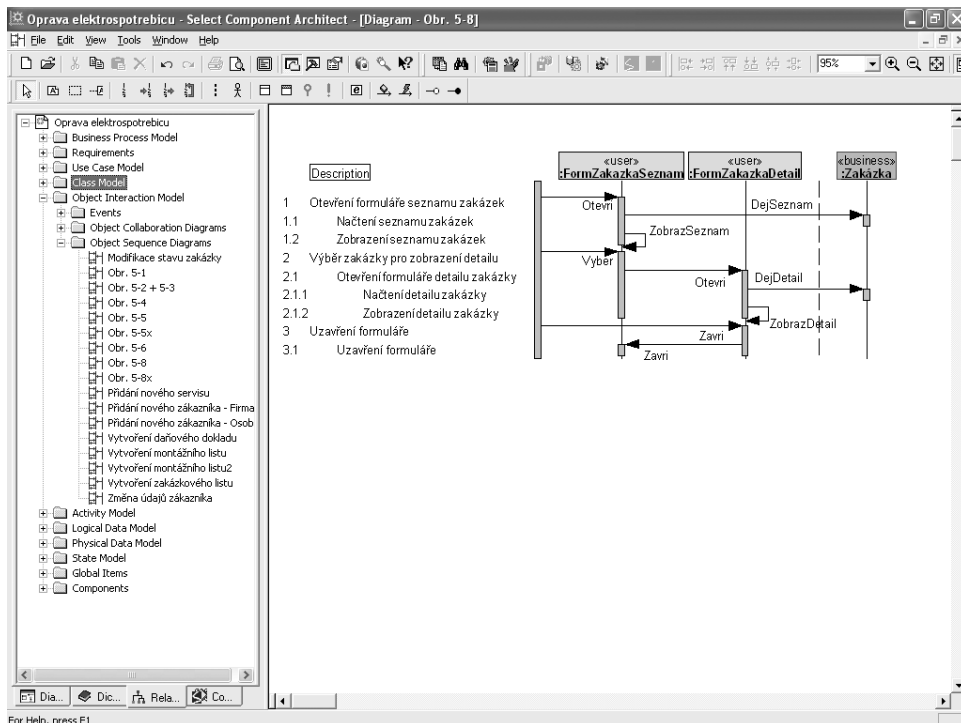
Diagramy interakce modelují realizace případů užití.

Existují dva typy diagramů interakce, sekvenční diagramy a diagramy objektové komunikace.

Diagramy objektové komunikace kladou důraz na spolupráci objektů, napovídají o struktuře balíčků, nejsou však příliš vhodné pro zachycení časové posloupnosti.

Sekvenční diagramy znázorňují spolupráci objektů z hlediska času.

-  Oba typy diagramů je obtížné a pracné kreslit bez automatizované podpory pomocí nástrojů CASE.
-  Na projektu zpravidla není prakticky možné kreslit diagramy pro všechny případy užití, vyberte proto nejvíce přínosné případy užití a jakési zástupce podobných skupin reprezentující vzory řešení.



Obrázek 6.11 Select Component Architect – sekvenční diagramy