

Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky

Katedra informačních technologií

Student : **Iveta Králová**
Vedoucí bakalářské práce : **doc. Ing. Alena Buchalcegová, Ph.D.**
Oponent bakalářské práce : **Ing. Jan Mészáros**

TÉMA BAKALÁŘSKÉ PRÁCE

UKÁZKOVÝ PŘÍKLAD OBJEKTOVÉ ANALÝZY
A NÁVRHU V CASE NÁSTROJI ASTAH*

ROK : 2011

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně a že jsem uvedla všechny použité prameny a literaturu, ze kterých jsem čerpala.

V Praze dne 30.4.2011

.....
podpis

Poděkování

Ráda bych tímto poděkovala vedoucí bakalářské práce, paní doc. Ing. Aleně Buchalceové, Ph.D. za vstřícnost, cenné rady a pečlivou korekturu celé práce.

Abstrakt

Bakalářská práce je zaměřena na vytváření diagramů jazyka UML 2.0 v CASE nástroji Astah* a je koncipována jako příručka pro studenty kurzu softwarového inženýrství.

Hlavním cílem práce je na praktickém příkladě ukázat použití vybraných diagramů UML ve fázi analýzy a návrhu informačního systému. Dalším cílem, který přímo vede k dosažení cíle hlavního, je vysvětlit logiku těchto UML diagramů a popsat jednotlivé kroky postupu jejich tvorby v CASE nástroji Astah*. Práce si klade za cíl také stručně charakterizovat CASE nástroj Astah* a popsat základní funkce využívané při práci v tomto nástroji.

Těchto cílů bylo dosaženo zejména studiem dostupných teoretických poznatků, využitím referenčního manuálu z nápovědy a tutoriálu ze stránek Astah* a v neposlední řadě také zpracováním informací z internetu.

Vlastním přínosem práce je zejména vytvoření jednoduchých a snadno pochopitelných UML modelů, které dokumentují konkrétní aplikaci. Dalším přínosem práce je zpracování návodů pro práci v programu Astah* edice professional, ve kterých je na rozdíl od obsáhlých manuálů popsána jen ta funkčnost programu, kterou uživatel nejvíce upotřebí.

Práce je rozdělena do dvou základních částí. V První části je nejprve vysvětlen pojem CASE nástroj. Následně je uvedena základní charakteristika CASE nástroje Astah* a popis základů práce v Astah* professional. Druhá část se soustředí na postup vytváření UML diagramů v Astah* professional. Nejprve definuje praktický příklad, který je následně namodelován pomocí jednotlivých UML diagramů v pořadí diagram případů užití, diagram tříd, diagram aktivit a sekvenční diagram.

Klíčová slova:

UML, modelovací jazyk, diagram, model, analýza a návrh, CASE nástroj

Abstract

The thesis is focused on creating UML 2.0 diagrams in CASE tool Astah* and it's conceived as a handbook for students of software engineering course.

The main aim of this thesis is to show a practical example of the use of selected UML diagrams in the analysis and design of information systems. Next aim that leads directly to achieving the main one is to explain the logical structure of the UML diagrams and describe the particular steps of their creation in CASE tool Astah *. The aim of the thesis is also to briefly characterize a CASE tool Astah * and describe the basic functions used to work in this tool.

The aims of this thesis were reached mainly by studying available theoretical findings, using the reference manual from the help and tutorial from Astah* web sites and last but not least, by information processing from the Internet.

The contribution of this thesis is especially a creation of simple and understandable UML models which document the particular application. Another contribution of the thesis are guidelines for work in the Astah *, edition professional, which describes only the functionality that user will actually use in contrast to the extensive manual.

The thesis is divided into two main parts. The first part explains the term of CASE tool initially. This part follows on a basic characteristic of the CASE tool Astah * and a description of the basic work in Astah * professional. The second part is focused on the creation of UML diagrams in Astah * professional. At first, there is defined a practical example, which is then modeled using particular UML diagrams in order of use case diagram, class diagram, activity diagram and sequence diagram.

Key words:

UML, modelling language, diagram, model, analysis and design, CASE tool

Obsah

1.	ÚVOD	8
1.1	Vymezení tématu práce a důvod výběru tématu	8
1.2	Cíle práce a smysl cílů	8
1.3	Způsob dosažení cílů	8
1.4	Předpoklady a omezení práce.....	8
1.5	Struktura práce	8
2.	SEZNÁMENÍ S CASE NÁSTROJEM ASTAH*	10
2.1	Základní charakteristika nástroje Astah*	10
2.2	Základy práce v nástroji Astah* professional	12
2.2.1	Instalace.....	13
2.2.2	První spuštění	13
2.2.3	Nový projekt a popis rozvržení obrazovky	14
2.2.4	Prvek modelu versus prvek diagramu - mazání, kopírování, vkládání	15
2.2.5	Možnosti zobrazení.....	16
2.2.6	Kopírování, vkládání, označování, úprava a přesun prvků diagramu	16
2.2.7	Export obrázku	17
2.2.8	Možnosti tisku	17
2.2.9	Editace prvků diagramu – viditelnost, změna velikosti, barva, notace	18
3.	POSTUP VYTVÁŘENÍ UML DIAGRAMŮ V ASTAH* PROFESSIONAL	22
3.1	Zadání aplikace OB ZAVODY	22
3.1.1	Popis požadavků na aplikaci OB ZAVODY	22
3.1.2	Slovníček pojmů	25
3.2	Diagram případů užití	26
3.2.1	Model případů užití aplikace OB ZAVODY	26
3.2.2	Postup tvorby diagramu případů užití v Astah* professional.....	28
3.2.3	Ukázková specifikace případu užití Přihlásit se na závod.....	30
3.3	Diagram tříd	33
3.3.1	Model tříd aplikace OB ZAVODY	35
3.3.2	Postup tvorby diagramu tříd v Astah* professional.....	35
3.4	Diagram aktivit.....	39
3.4.1	Diagram aktivit aplikace OB ZAVODY	39
3.4.2	Postup tvorby diagramu aktivit v Astah* professional	41
3.5	Sekvenční diagram	43
3.5.1	Sekvenční diagram aplikace OB ZAVODY	43
3.5.2	Postup tvorby sekvenčního diagramu v Astah* professional.....	44
4.	ZÁVĚR.....	46
5.	SEZNAM POUŽITÉ LITERATURY	47

5.1	Knižní zdroje	47
5.2	Internetové zdroje.....	47
6.	TERMINOLOGICKÝ SLOVNÍK	49

1. ÚVOD

1.1 Vymezení tématu práce a důvod výběru tématu

Bakalářská práce je zaměřena na vytváření diagramů jazyka UML 2.0 v CASE nástroji Astah* a je koncipována jako příručka pro studenty kurzu softwarového inženýrství. Na praktickém příkladě inspirovaném reálně využívanou aplikací je vysvětlena logika jednotlivých UML diagramů a následně popsán postup jejich vytváření v CASE nástroji Astah*.

Důvodem výběru tématu byla možnost získat praktické zkušenosti v oblasti analýzy a návrhu aplikací a zároveň obohatit skupinu studijních materiálů pro výuku kurzu softwarového inženýrství.

1.2 Cíle práce a smysl cílů

Hlavním cílem bakalářské práce je na praktickém příkladě ukázat použití vybraných diagramů UML ve fázi analýzy a návrhu informačního systému.

Dalším cílem, který přímo vede k dosažení cíle hlavního, je vysvětlit logiku těchto UML diagramů a popsat jednotlivé kroky postupu jejich tvorby v CASE nástroji Astah*.

Dále si práce klade za cíl také stručně charakterizovat CASE nástroj Astah* a popsat základní funkce využívané při práci v tomto nástroji.

Výsledky bakalářské práce budou publikovány na internetu. K dispozici tak budou nejen studentům, ale i ostatním zájemcům o problematiku modelování v UML.

1.3 Způsob dosažení cílů

Hlavního cíle práce bylo dosaženo studiem dostupné literatury, věnující se problematice modelování v UML.

Pro popis základů práce v nástroji Astah* a postupu tvorby UML diagramů v tomto nástroji byl využit referenční manuál z nápovědy, tutoriál stažený ze stránek Astah* a hlavně vlastní zkušenost s používáním programu.

Charakteristika CASE nástroje Astah* byla zpracována z informací získaných prostřednictvím internetových stránek.

1.4 Předpoklady a omezení práce

Zadání praktického příkladu je inspirováno reálně používanou aplikací, která je v praxi velmi obsáhlá a vzhledem k rozsahu práce ji není možné namodelovat kompletně. Jelikož příklad ukázkové analýzy a návrhu v UML má sloužit především jako studijní materiál, z reálně používané aplikace byl vybrán pouze jeden modul, který je snadno pochopitelný a v některých funkcích a vlastnostech se od reálné aplikace odlišuje.

Práce je zaměřena výhradně na tvorbou UML diagramů na konceptuální úrovni, což je vzhledem k vysvětlení logiky diagramů a popisu kroků postupu jejich tvorby zcela postačující.

Jednotlivé UML diagramy, popisy jejich tvorby i popis základů práce v nástroji Astah* jsou vytvořeny ve verzi 6.3 nástroje Astah*, edice professional. Při dokončování bakalářské práce Astah* uvedl na trh novou verzi 6.4, která se od předchozí příliš neliší a její použití by tak na žádný z cílů práce nemělo vliv.

1.5 Struktura práce

Bakalářská práce je rozdělena do dvou základních částí (druhá a třetí kapitola práce). Vzhledem k tomu, že práce má sloužit jako příručka, obě její části jsou zaměřeny především prakticky.

Úkolem druhé kapitoly práce je seznámení s CASE nástrojem Astah*. V úvodu kapitoly je nejprve vysvětlen pojem CASE nástroj. V první podkapitole je potom uvedena základní charakteristika CASE nástroje Astah*, včetně jeho dostupných edicí. Druhá podkapitola popisuje základy práce v Astah*, edice professional.

Třetí kapitola práce se soustředí na postup vytváření UML diagramů v Astah* professional. V první podkapitole je nejprve definován praktický příklad. Následující podkapitoly se potom věnují modelování jednotlivých UML diagramů, které dokumentují zadaný příklad, v pořadí diagram případů užití, diagram tříd, diagram aktivit a sekvenční diagram.

2. SEZNÁMENÍ S CASE NÁSTROJEM ASTAH*

V této kapitole se v první části zaměřuji na základní charakteristiku CASE nástroje Astah*, tzn., k čemu nástroj slouží a jaké jsou jeho dostupné edice. Druhá část je zaměřena na základy práce v nástroji Astah*, edice professional a jejím úkolem je pomoci uživatelům hlavně při orientaci a prvních krocích v programu.

Ještě předtím, než se budu zabývat přímo CASE nástrojem Astah*, je účelné vymezit samotný pojem CASE nástroj a možnosti jeho využití. Zkratka CASE je označením pro Computer Aided Software Engineering, což v přesném překladu znamená počítačem podporované softwarové inženýrství [19], lépe řečeno „nástroje pro podporu analýzy a návrhu aplikací“ [6].

Při analýze a návrhu rozsáhlejších aplikací, obzvláště pak objektově orientovaných, se v současnosti obvykle používá modelovací jazyk UML (Unified Modeling Language). Jazyk UML je „souhrnem především grafických notací k vyjádření analytických a návrhových modelů“ [6] a právě z něj všechny tyto CASE nástroje vychází. Je samozřejmě možné kreslit UML diagramy i v kreslicím nástroji, nebo na papíře, ale CASE nástroj proces návrhu dokáže usnadnit a zefektivnit díky mnohem rozšířenějším možnostem oproti kreslicímu nástroji. CASE nástroje zpravidla umožňují:

- modelování informačního systému pomocí diagramů [19],
- generování zdrojového kódu z modelu, což usnadňuje následné programování [19],
- zpětné vytvoření modelu podle existujícího zdrojového kódu (reverse engineering) [19],
- synchronizaci modelu a zdrojového kódu [19],
- vytvoření dokumentace z modelu [19],
- propojení jednotlivých modelů a zachycení celého analytického návrhu ve společné repository sdílené týmy vývojářů [6],
- procesní modelování jako doplněk k UML technikám [6].

CASE nástrojů je na trhu celá řada a mohou poskytovat různě rozsáhlou funkcionalitu, které potom odpovídají i různé cenové kategorie. Z běžně používaných nástrojů mohu kromě Astah* jmenovat například Enterprise Architect firmy Sparx Systems, MagicDraw firmy No Magic, Select Component Architect firmy Select Business Solutions, PowerDesigner firmy Sybase, AllFussion firmy Computer Associates, Oracle Designer firmy Oracle a další.

2.1 Základní charakteristika nástroje Astah*

Astah* je CASE nástroj vytvořený pro modelování v jazyce UML, jehož tvůrcem je japonská firma ChangeVision. [17] Jeho původní název byl JUDE, ale protože na tento název byly negativní ohlasy ze strany německých uživatelů (pozn.: JUDE znamená v němčině Žid.), došlo v září roku 2009 k jeho přejmenování na Astah*. [16]

Kromě samotného UML modelování Astah* nabízí i další prvky pro vývojáře softwaru, např. diagram vztahů mezi entitami (ERD = ER diagram), CRUD matici, vývojový diagram (FlowChart), myšlenkovou mapu (Mind Map), nebo diagram toků dat (DFD = Data Flow Diagram). [9]

Astah* také zefektivňuje komunikaci mezi vývojáři, analytiky, testery a manažery, jelikož všechny UML diagramy jsou v nástroji důsledně ukládány do jednoho modelu. [9]

V současnosti nejnovější verze produktu je 6.4.

Astah* je vydáván ve čtyřech edicích (+ Astah* pro modelování na IPADU) podle toho, jak širokou funkcionalitu edice poskytuje a jaká licence je k ní potřebná: [9]

1. **Astah* professional**, což je kompletní edice poskytující následující funkcionalitu:
→ Diagramy, jejichž výčet a stručný popis je zapsán v tabulce 2.1.

Název diagramu:	Popis diagramu:
UML diagramy verzí 2.x: • Diagram tříd (Class diagram), • Diagram případů užití (UseCase diagram), • Stavový diagram (Statemachine diagram), • Diagram aktivit (Activity diagram), • Sekvenční diagram (Sequence diagram), • Diagram komunikace (Communication diagram), • Diagram komponent (Component diagram), • Diagram nasazení (Deployment diagram), • Diagram vnitřní struktury (Composite Structure diagram), • Objektový diagram (Object Diagram).	UML diagramy verzí 2.x jsou oproti předchozím verzím UML 1.x rozšířeny o nové typy diagramů : Diagram přehledu interakcí, diagram načasování a diagram vnitřní struktury. Diagramy spolupráce byly ve verzích 2.x přejmenovány na diagramy komunikace. Dále byly do UML diagramů 2.x také oficiálně zahrnuty diagramy objektů a diagramy balíčků, které se v praxi již běžně používaly. [6] Nalevo je uveden seznam diagramů, které podporuje Astah* professional.
Myšlenkové mapy (Mind Map *1)	Účinný způsob, jak vizuálně zachytit své myšlenky, cíle a nápady. Mohou být využity pro záznamy z brainstormingu, při zachycování požadavků uživatele nebo také pro zápis poznámek ze schůze. V kapitole 2.2.2 je odkázáno na tři ukázky myšlenkových map z nápovědy.
Diagram vztahů mezi entitami (ER Diagram = Entity Relationship diagram)	Zachycuje objekty reality (entity) a jejich vzájemné vztahy při návrhu datové základny.
Vývojový diagram (Flowchart)	Využívá se pro znázornění nějakého procesu nebo algoritmu. Flowchart v Astah* nabízí celou řadu různých symbolů, které se propojují pomocí šipek, a lze tak zachytit téměř jakýkoliv postup. Tento typ diagramu je hojně využíván hlavně při procesním modelování.
CRUD matice (CRUD)	Tabulka, která slouží ke zjištění kompatibility mezi funkcionalitou, vyjádřenou pomocí případů užití a datovým modelem, zachyceným v ER diagramu. Sloupce tabulky představují entity a řádky tabulky představují případy užití. Do polí tabulky se zapisují písmena C-Create, R-Read, U-Update, D-Delete (možno v jednom poli i všechna najednou) podle toho, zda se entita v daném případě užití vytváří, čte, aktualizuje nebo maže.
Diagram toků dat (DFD = Data flow Diagram)	Slouží ke znázornění toků dat mezi systémy.

Tabulka požadavků (Requirement Table)	Je vhodná pro zaznamenání požadavků pro daný projekt. Jednotlivé požadavky mohou být sestaveny hierarchicky a skládají se z ID, názvu a textu.
Diagram požadavků (Requirement Diagram)	Způsob vizuálního zachycení hierarchie požadavků pro daný projekt a jejich vzájemných vztahů.

Tabulka 2.1 Přehled diagramů edice Astah* professional, zdroj: autor dle [11]

→ Podpora exportu navržených modelů do zdrojového kódu a zpětné vytváření modelů z existujícího zdrojového kódu (reverzní engineering) pro tyto programovací jazyky:

- Java,
- C# ,
- C++. [11]

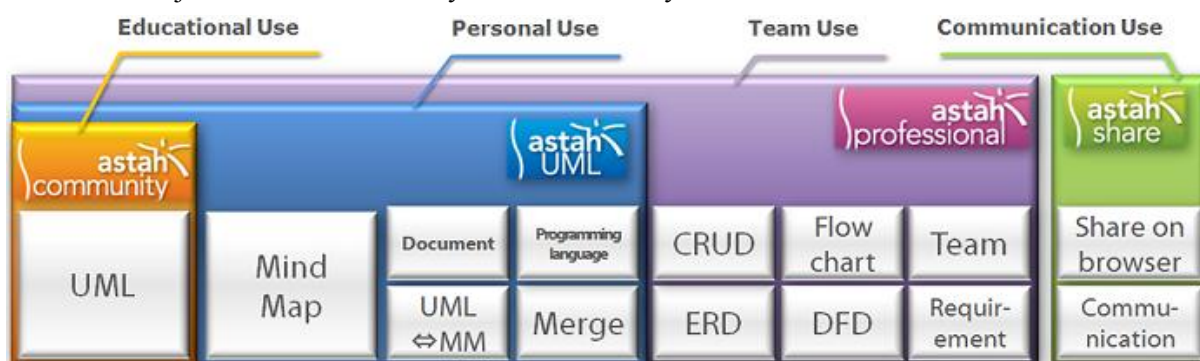
2. **Astah* UML**, která oproti Astah* professional zahrnuje pouze diagramy UML2.x (ve stejné šíři jako Astah* professional) a Mind Map *1. [13]

3. **Astah* Community**, což je verze „nejodlehčenější“, zdarma a poskytuje čistě jen všechny diagramy UML2.x. [10]

4. **Astah* Share**, což je komunikační nástroj, který umožňuje sdílet Astah* soubory s ostatními. Každý člen vývojového týmu může prostřednictvím webového prohlížeče přistupovat ke sdílenému obsahu a komentovat jej. Podporuje také diskuze o nápadech a je tak efektivní při spolupráci více návrhářů, analytiků, či vývojářů. [12]

⊕ **Astah* UML** pro modelování UML na IPADU. Je zdarma a umožňuje snadno a rychle modelovat UML diagramy tříd (Class diagrams), včetně jejich exportu do PNG a XML. [14]

Přehledně jsou možnosti těchto čtyř edicí znázorněny na obrázku 2.1.



Obrázek 2.1 Přehled Astah edicí verze 6.4, zdroj: [9]

Zakoupit licenci je nutné pouze k edicím Astah* professional, Astah* UML a Astah* Share. Edice Astah* Community i Astah* UML pro IPAD jsou zdarma.

Podmínky licence pro Astah* professional na Fakultě informatiky a statistiky VŠE v Praze a soubor s licenčním klíčem jsou dostupné po přihlášení na webu <https://kitscm.vse.cz/>, viz kapitola 2.2.1.

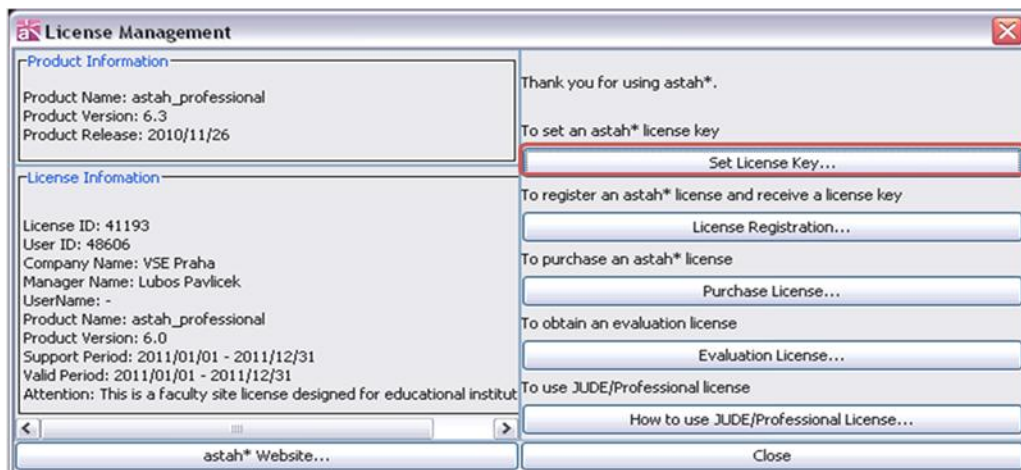
2.2 Základy práce v nástroji Astah* professional

V následujících podkapitolách jsou popsány základy práce v nástroji. Cílem není vyčerpávajícím způsobem popsat funkcionalitu programu, ale spíše vystihnout to hlavní, co skutečně, hlavně začínající, uživatel upotřebí. Záměrně se v této kapitole nezabývám tvorbou jednotlivých typů diagramů, která je předmětem třetí kapitoly této práce. Pro zpracování kapitoly byly použity zdroje [8] a [24].

2.2.1 Instalace

Postup instalace je popsán na stránkách <https://kitscm.vse.cz/> , ale je dostupný jen po přihlášení, proto jej zde uvádím:

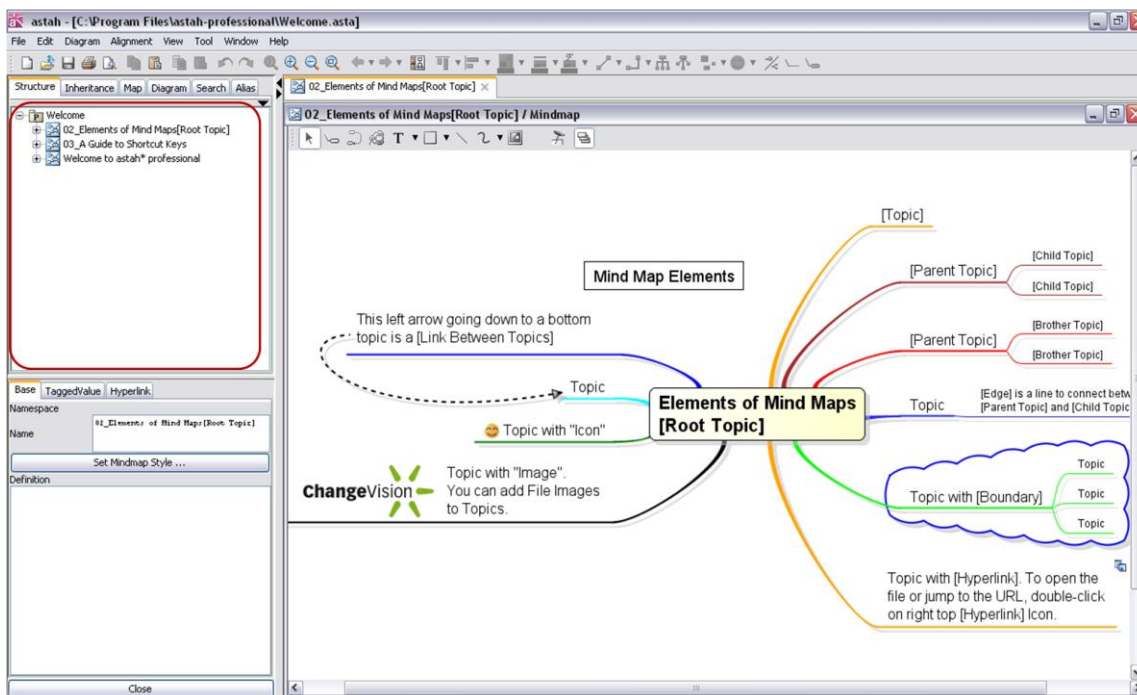
1. Ze stránky <http://astah.change-vision.com/en/product/astah-professional.html> si stáhněte poslední verzi produktu Astah* professional a tu nainstalujte.
2. Pokud máte možnost přihlášení na <https://kitscm.vse.cz/>, v levém menu v sekci Informace klikněte na odkaz [*Licence astah**]. Ve spodní části stránky naleznete licenční soubor [astah_professional_license.xml](#), který si stáhněte.
3. Spustěte program Astah* professional. Při spuštění se Vám objeví dotaz na zadání licenčního souboru jako na obrázku 2.2. Zvolte [*Set License Key*] a vyberte stažený licenční soubor. [23]



Obrázek 2.2 Výběr licenčního souboru, zdroj: prostředí programu Astah* professional dle [23]

2.2.2 První spuštění

Po dokončení instalace se objeví úvodní obrazovka, jejíž ústřední část tvoří černá plocha. Levý panel se složkou *Welcome*, jako na obrázku 2.3, vyvoláte přes hlavní nabídku [*Help*] – [*Welcome to Astah* professional*].



Obrázek 2.3 Uvítací obrazovka, zdroj: prostředí programu Astah* professional

Na panelu *Welcome* je aktivní záložka *Structure*, pod níž si můžete prohlédnout tři ukázky motivačních myšlenkových map (Mind Map), vztahujících se k samotnému programu:

- *02_Elements of Mind Maps [Root Topic]* - mapa prvků myšlenkových map,
- *03_A Guide to Shortcut Keys* - mapa klávesových zkratk programu,
- *Welcome to Astah* professional* - mapa toho, co nabízí Astah* professional.

2.2.3 Nový projekt a popis rozvržení obrazovky

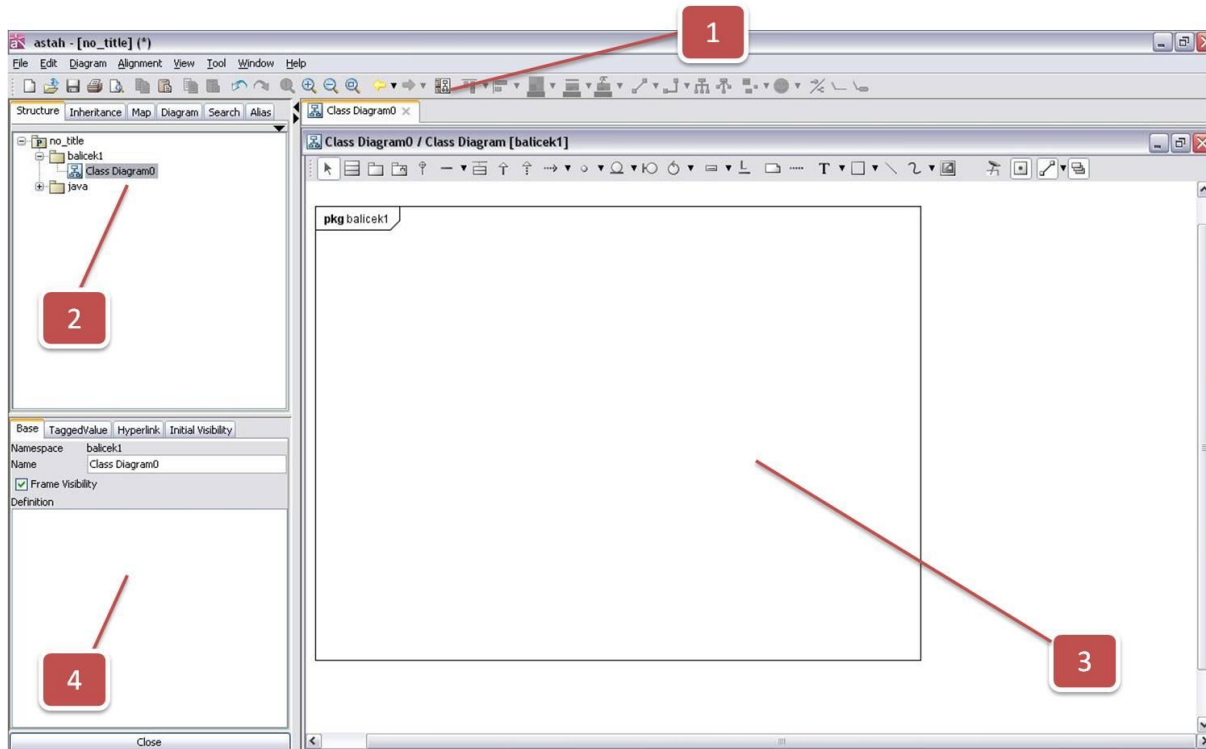
Pro demonstraci pracovního rozvržení obrazovky programu si nejprve založte nový projekt¹:

1. V hlavní nabídce vyberte [*File*] – [*New*], nebo můžete zvolit ikonu bílého listu [*Create new file*] vlevo nahoře na nástrojové liště.
2. Nový projekt je založen s výchozím názvem *no_title* a s jedním předpřipraveným balíčkem *java*. Změnit jméno projektu můžete při ukládání.

Do projektu nyní můžete přidávat jednotlivé diagramy, které ve výsledku budou tvořit model navrhované aplikace. Všechny tyto diagramy mohou být vytvořeny jako součást balíčků. Následuje postup vytvoření balíčku a do něj vložení diagramu tříd:

1. Pravým tlačítkem myši klikněte na složku projektu (výchozí *no_title*, nebo Vámi přiřazený název při ukládání), vyberte [*Create Model*] – [*Add Package*] a pojmenujte si balíček, např. *balicek1*.
2. Dvě alternativy vložení diagramu tříd do balíčku:
 - a) Pravým tlačítkem myši klikněte na balíček *balicek1* a vyberte [*Create Diagram*] – [*Add Class Diagram*].
 - b) Označte balíček a v hlavní nabídce vyberte [*Diagram*] – [*Class Diagram*].

Po vytvoření diagramu tříd bude prostředí vypadat tak, jak je znázorněno na obrázku 2.4.



Obrázek 2.4 Rozvržení obrazovky při práci, zdroj: prostředí programu Astah* professional

¹ Pro úplnost, nový projekt je také možné vytvořit ze šablony – v hlavní nabídce vyberete [*File*] – [*New by Template*].

Následuje stručný popis jednotlivých částí obrazovky, očíslovaných na obrázku 2.4.

1. **Management view** se skládá z hlavního menu a nástrojové lišty. Hlavní menu obsahuje funkce vztahující se k celému projektu jako je např. práce se souborem, editace, zobrazení. Na nástrojové liště jsou umístěny nejčastěji využívané funkce.
2. **Project view** poskytuje celkový přehled projektu. Můžete zde přepínat mezi záložkami *Structure*, *Inheritance*, *Map*, *Diagram*, *Switch* a *Alias*. V tabulce 2.2 je popsáno, k čemu tyto záložky slouží.

Záložka Structure	Zobrazuje modely pomocí stromovité struktury, kdy je možné některé operace s prvky ve struktuře vyvolat prostřednictvím lokálního menu.
Záložka Inheritance	Zobrazuje modely ve stromovité struktuře na základě vztahu dědičnosti mezi třídami.
Záložka Map	Pohled na diagram, který je právě otevřený v editoru diagramu. Modře vybarvená oblast značí, která část diagramu je právě vidět v editoru. Posunout ji můžete tažením levým tlačítkem myši a změnit její velikost můžete tažením pravým tlačítkem myši, což je velmi užitečné při tvorbě rozsáhlých diagramů.
Záložka Diagram	Seznam všech diagramů v projektu. Výběrem z tohoto seznamu je možné diagram zobrazit v editoru diagramu.
Záložka Switch	Slouží k vyhledávání prvků modelu nebo k nahrazování řetězců v názvech prvků modelu a neplatných odkazech.
Záložka Alias	Pohled, který slouží k nastavení aliasů prvků modelu v projektu.

Tabulka 2.2 Záložky v Project view, zdroj: autor

3. **Editor diagramu** představuje stěžejní část okna, kde se vytváří a upravují jednotlivé diagramy a modely. Můžete mít otevřeno i několik diagramů, mezi nimiž lze přepínat pomocí záložek.
4. **Property view** zobrazuje vlastnosti jednotlivých prvků modelu a umožňuje jejich editaci. Jeho podoba je závislá na právě vybraném prvku modelu a pro každý typ prvku modelu je odlišná.

2.2.4 Prvek modelu versus prvek diagramu - mazání, kopírování, vkládání

Pod pojmem prvek v projektu, ať už prvek modelu či diagramu, je možné si představit např. třídu, případ užití, aktéra, atd. Zásadní rozdíl mezi prvkem modelu a prvkem diagramu je, že prvek modelu existuje někde v projektu, v jakémkoli diagramu (či ani v žádném diagramu být nemusí), kdežto prvek diagramu je pouze grafické znázornění, notace daného prvku modelu v nějakém diagramu. Stejný prvek modelu tak může být zastoupen i ve více diagramech jako jejich prvek, přičemž v každém diagramu může vypadat jinak (jiná barva, velikost, styl písma, atd.). Názorně tuto situaci vystihuje obrázek 2.5.

Pokud chcete **vymazat** prvek pouze **z diagramu**:

1. klikněte na něj v editoru diagramu pravým tlačítkem myši,
2. zvolte [*Delete from Diagram*].

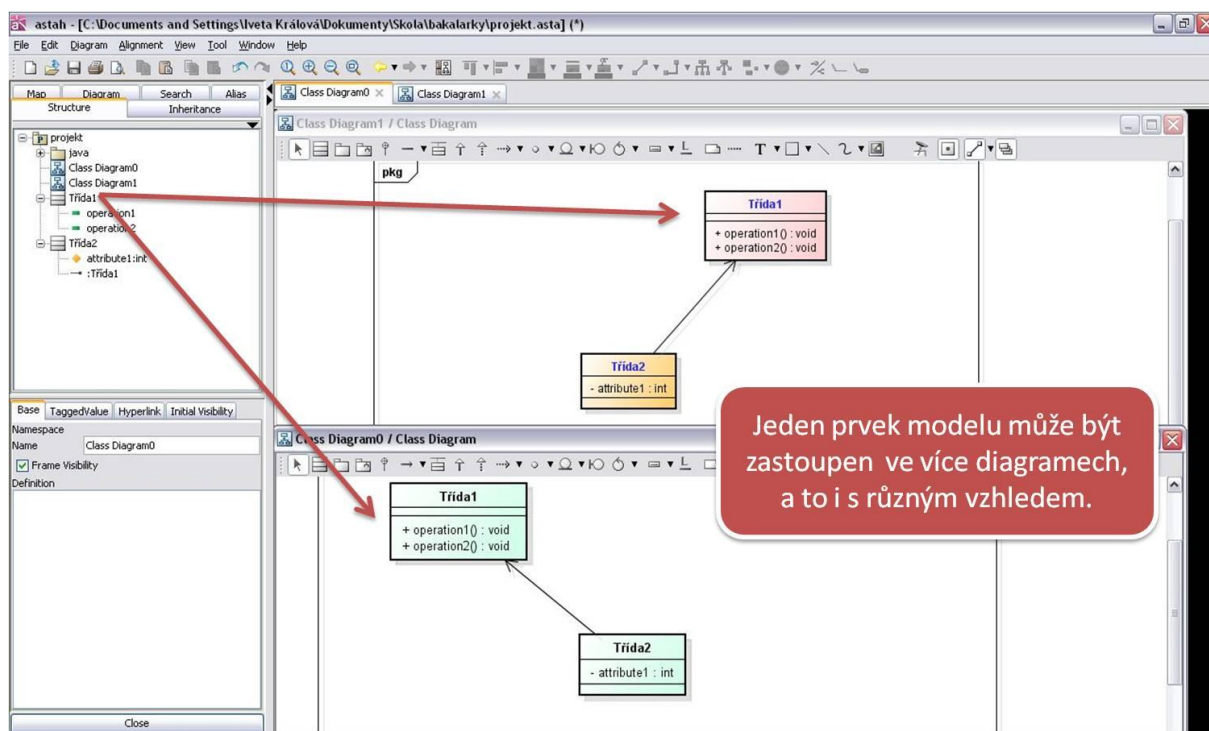
Prvek se z diagramu odstraní, ale v modelu zůstane.

Pokud chcete **vymazat** prvek z modelu a tedy i celého **projektu**:

1. klikněte na něj v editoru diagramu pravým tlačítkem myši,
2. zvolte [*Delete from Model*].

Při **kopírování** a následném **vkládání** prvků se chování prvků liší v závislosti na typu diagramu:

- a) u diagramů tříd (Class diagram), diagramů případů užití (UseCase diagram), ER diagramů a diagramů toků dat (Data flow diagram) se kopírují a vkládají pouze prvky diagramu, ale nikoli prvky modelu,
- b) u ostatních typů diagramů se kopírují a vkládají zároveň prvky diagramu i prvky modelu.



Obrázek 2.5 Jeden prvek modelu ve více diagramech, zdroj: prostředí programu Astah* professional

2.2.5 Možnosti zobrazení

Pro změnu zobrazení editoru diagramu je možno využít buď hlavní nabídku – [View], nebo přímo ikony na panelu nástrojů.

Možnosti změny zobrazení jsou popsány v tabulce 2.3:

[Zoom]	Obnoví výchozí velikost diagramu (100%) v editoru diagramu.
[Zoom in]	Zvětší diagram.
[Zoom out]	Zmenší diagram.
[Fit in window]	Zobrazí pohled na celý diagram.
[Back to previous editor]	Zobrazí předchozí diagram.
[Forward to next editor]	Zobrazí následující diagram.
[Show/hide project view]	Slouží k zobrazení nebo skrytí položek v project view.
[Look & Feel]	Zde je možné nastavit si požadovaný vzhled programu (Metal, Nimbus, CDE/Motif, Windows, Windows Classic).
[Alias]	Volba zobrazení jména prvků nebo jejich aliasů.

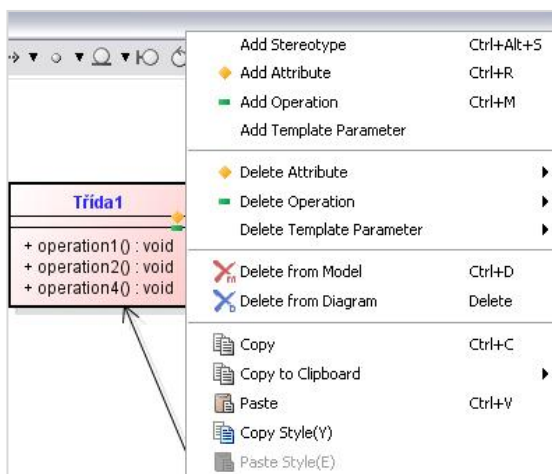
Tabulka 2.3 Možnosti změny zobrazení, zdroj: autor

2.2.6 Kopírování, vkládání, označování, úprava a přesun prvků diagramu

Kopírovat a vkládat je možné buď celé prvky diagramu, nebo jen jejich styl (barva pozadí, styl čáry, barva písma, font písma, atd.). Z lokálního menu (obrázek 2.6) prvku diagramu si vyberete podle potřeby:

- [Copy]/[Paste] nebo [Ctrl + C]/[Ctrl + V] pro kopírování a vkládání celého prvku diagramu,

b) [Copy Style]/[Paste Style] pro kopírování a vkládání pouze stylu prvku diagramu.



Obrázek 2.6 Kopírování a vkládání, zdroj: prostředí programu Astah* professional

Označování prvků diagramu se provádí standardně kliknutím levým tlačítkem myši. Pokud chcete označit více prvků najednou, s kliknutím podržte [Shift]/[Ctrl]. Všechny prvky v diagramu se označí pomocí [Ctrl + A].

Úprava textu v prvku diagramu je intuitivní. Stiskněte klávesu [F2], nebo na prvek dvakrát klikněte a můžete text přímo editovat. Pokud chcete změnit pořadí atributů, jednoduše je myší přetáhněte.

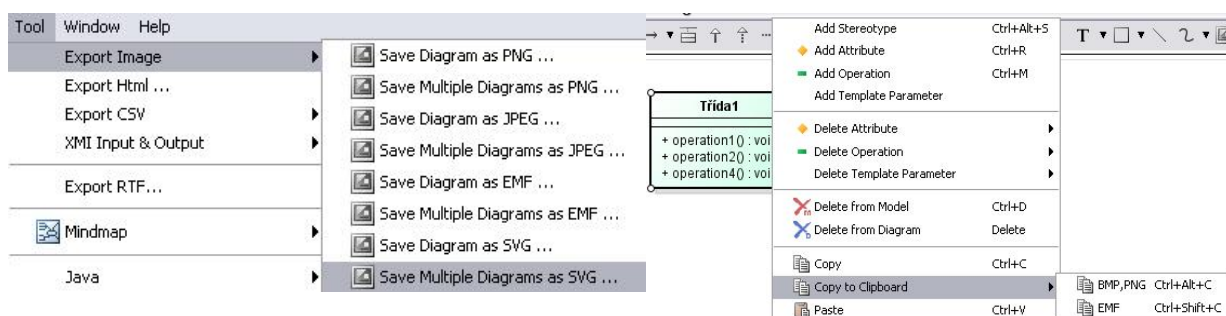
Prvky se **přesouvají** tažením levým tlačítkem myši nebo pomocí šipek. Pro vertikální/horizontální přesun spolu s levou myší podržte [Shift] a pro přesun po mřížce podržte [Ctrl].

2.2.7 Export obrázku

Z diagramů projektu ve formátu *asta* je možné získat grafický soubor ve formátech JPEG, PNG, SVG nebo EMF:

- V hlavním menu zvolte [Tool] – [Export Image], kde vyberete, do jakého formátu chcete exportovat a zda jen jeden, nebo více diagramů.
- V lokálním menu u prvku diagramu vyberte [Copy to Clipboard].

Možnosti exportu obrázku jsou znázorněny na obrázku 2.7.



Obrázek 2.7 Export obrázku, zdroj: prostředí programu Astah* professional

2.2.8 Možnosti tisku

Vše potřebné k tisku projektu a diagramů je umístěno v hlavní nabídce - [File]. Jednotlivé možnosti tisku jsou popsány v tabulce 2.4:

[Print Setup (Project)]	Nastavení vlastností pro tisk celého projektu, tedy všech diagramů. Dají se zde nastavit např. vlastnosti stránky (velikost papíru, orientace stránky, okraje), záhlaví a zápatí, vlastnosti pro všechny diagramy a další.
[Print Setup (Diagram)]	Nastavení vlastností pro tisk aktuálního diagramu v editoru diagramu. Zde je možné na záložce <i>Page</i> zatrhnout volbu <i>[Print by using the print setting for the project]</i> , která znamená, že vlastnosti tisku pro tento diagram budou ty, které byly nastaveny v <i>[Print Setup (Project)]</i> a nebo tuto volbu odškrtnout a nastavit jiné vlastnosti pro tento diagram, než je tomu u ostatních diagramů v projektu.
[Print Preview]	Zobrazí náhled tisku na aktuální diagram v editoru diagramu.
[Print]	Vytiskne aktuální diagram v editoru digramu.
[Preview multi]	Zobrazí náhled tisku na více diagramů. Jsou zde čtyři možnosti podle toho, na jaké typy diagramů chcete nahlédnout: ○ <i>[Print Preview Multi-diagram]</i>, ○ <i>[Print Preview Multi-UseCase Description]</i>, ○ <i>[Print Preview Multi-CRUD]</i>, ○ <i>[Print Preview Multi-Requirement Table]</i>. Podle zvolené možnosti se zobrazí tabulka s diagramy daného typu, kde zatrhnete diagramy k požadovanému náhledu.
[Print multi]	Vytiskne více diagramů, které se zvolí obdobně jako v <i>[Preview multi]</i> .

Tabulka 2.4 Volby při tisku, zdroj: autor

2.2.9 Editace prvků diagramu – viditelnost, změna velikosti, barva, notace

Nastavení viditelnosti jednotlivých částí prvků diagramu je užitečné zejména pro rozlišení různých úrovní vývoje softwaru. Na konceptuální úrovni model mapuje realitu bez implementačních detailů, a proto je vhodné vypnout viditelnost u některých nadbytečných charakteristik. Oproti tomu model na designové úrovni původní konceptuální model rozšiřuje a je tedy potřeba, aby jeho implementační charakteristiky byly viditelné.

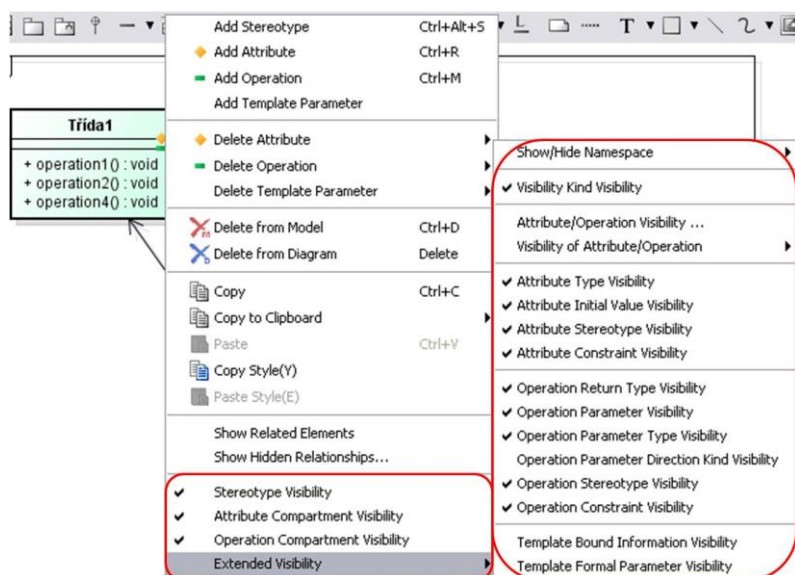
Viditelnost jednotlivých částí prvků diagramu na *on/off* je možné nastavit následujícími dvěma způsoby:

- a) Přímo v místní nabídce určitého prvku, která je závislá na typu prvku.

Například místní nabídka třídy zobrazí možnosti nastavení viditelnosti, jako je znázorněno na obrázku 2.8. Je zde možné například:

- vypnout/zapnout viditelnost atributů – *[Attribute Compartment Visibility]*,
- vypnout/zapnout viditelnost operací – *[Operation Compartment Visibility]*,
- vypnout/zapnout viditelnost *stereotypů* *[Stereotype Visibility]*,
- rozbalit přes *[Extended Visibility]* další možnosti viditelnosti, kde je možné vypnout/zapnout viditelnost jednotlivých částí atributů a operací (např. Inicializační hodnotu u atributů, návratovou hodnotu u operací, atd.).

- b) V hlavní nabídce nastavit výchozí viditelnosti, které budou potom platit pro všechny prvky: *[Tool]* – *[System Properties]* – *[Visibility 1/2/3 (initial)]*.



Obrázek 2.8 Viditelnost v místní nabídce třídy, zdroj: prostředí programu Astah* professional

Změna velikosti prvků diagramu se provádí:

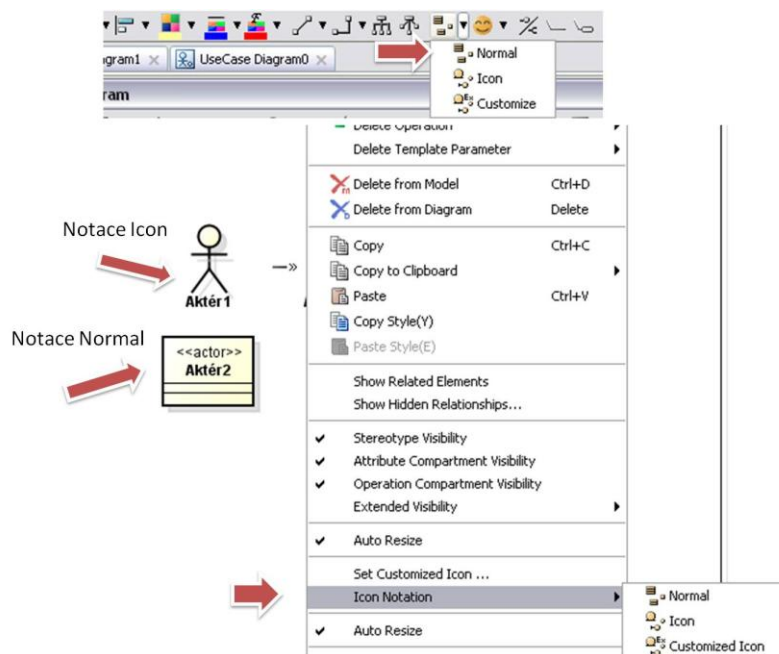
- Automaticky – velikost prvku se přizpůsobuje množství textu uvnitř něj.
- Tažením za roh prvku – velikost prvku je možno upravit podle potřeby, přičemž se ale vypne automatická velikost

Pozn.: Pokud chcete automatickou velikost potom znovu zapnout, je zde tato možnost v místní nabídce prvku - [Auto Resize].

Co se týče textu uvnitř nějakého prvku a jeho rozmístění do řádků, je zpravidla závislé na šířce prvku. Pokud ovšem z nějakého důvodu potřebujete řádek ukončit, stiskněte [Enter+Shift/Ctrl/Alt].

Pokud nejste spokojeni s výchozí **notací** prvků vkládaných do diagramu, můžete ji snadno změnit.

Změnu notace u vybraného prvku diagramu, je možné provést buď na nástrojové liště, nebo v lokálním menu prvku. Obrázek 2.9 znázorňuje možnost změny notace na příkladě aktéra.



Obrázek 2.9 Možnosti změny notace, zdroj: prostředí programu Astah* professional

Nastavení vhodné barvy prvků diagramu může přispět k přehlednosti diagramu. Je to dobrý způsob, jak některé prvky diagramu odlišit od ostatních, nebo naopak zdůraznit, že patří k sobě. Neměli byste ovšem zapomínat na čitelnost a použitelnost diagramu, proto doporučuji volit barvy v dostatečném kontrastu s textem uvnitř prvku.

Barvu právě vybraných prvků diagramu je možné změnit:

- z nástrojové lišty - [Set Color],
- z hlavního menu - [Edit] – [Set Color],
- z místní nabídky prvku diagramu - [Set Color].

Všechny tyto možnosti vyvolají dialogové okno pro výběr barvy jako na obrázku 2.10.

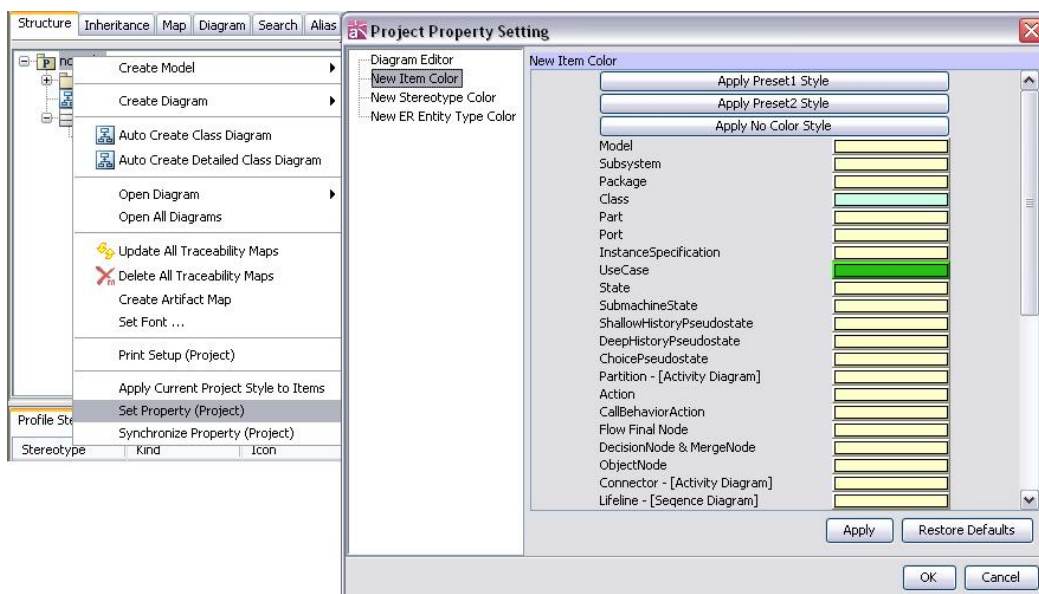


Obrázek 2.10 Výběr barvy, zdroj: prostředí programu Astah* professional

Nastavení výchozí barvy pro všechny nově přidané prvky v projektu můžete provést:

- v hlavním menu zvolte [Tool] – [Project] – [Set Project Properties...] – [New Item Color],
- v Project view na záložce [Structure] pravým tlačítkem myši rozbalte místní nabídku projektu (viz obrázek 2.11) a zvolte [Set Property (Project)] – [New Item Color].

Obě možnosti vyvolají dialogové okno pro nastavení výchozích barev nově přidaných prvků diagramu v daném projektu – obrázek 2.11, ale u stávajících prvků projektu se toto nastavení nijak neprojeví.



Obrázek 2.11 Nastavení barvy nově přidaných prvků diagramu v projektu, zdroj: prostředí programu Astah* professional

Kromě nastavení barev pro prvky přidané do projektu, je možné obdobně nastavit i výchozí barvy prvků v celém programu: V hlavním menu zvolte [Tool] – [System Properties] – [New Item Color].

Vyvolané dialogové okno je stejné jako u projektu. Rozdíl nastavení barev pro projekt a pro program je pouze v tom, že programové nastavení platí obecně, i v projektech, kde žádné barvy nadefinovány nebyly, kdežto projektové nastavení platí pouze pro jeden daný projekt. Pokud barvy prvků v projektu nastaveny jsou a barvy prvků v programu také, má přednost nastavení pro projekt.

Možnost synchronizace nastavení projektu s nastavením programu:

- a) v hlavním menu - [Tool] – [Project] – [Synchronize Project Properties...] – [OK],
- b) v Project view na záložce [Structure] pravým tlačítkem myši rozbalte místní nabídku projektu a zvolte [Synchronize Property (Project)] – [OK].

Funkcionalita programu je velmi obsáhlá, proto v této kapitole byly popsány jen základy práce v programu. Některé další užitečné funkce jsou zahrnuty v následující kapitole, která se zabývá tvorbou UML diagramů v programu. Kompletní a přehledně zpracovaný manuál naleznete v hlavní nabídce programu - [Help] – [astah* Reference Manual].

3. POSTUP VYTVÁŘENÍ UML DIAGRAMŮ V ASTAH* PROFESSIONAL

Cílem této kapitoly je na praktickém příkladě ukázat použití UML diagramů ve fázi analýzy a návrhu aplikace na konceptuální úrovni a popsat postup jejich tvorby v CASE nástroji Astah* professional.

V první části kapitoly je vymezeno zadání praktické aplikace s názvem OB ZAVODY, včetně popisu požadavků na aplikaci a slovníčku pojmů problémové oblasti.

Následující části kapitoly jsou zaměřeny na modelování jednotlivých UML diagramů: Diagram případů užití, diagram tříd, diagram aktivit a sekvenční diagram. Každá z těchto částí obsahuje stručnou obecnou charakteristiku diagramu, diagram aplikace OB ZAVODY spolu s objasněním jeho logické struktury a postupem jeho tvorby v Astah* professional.

3.1 Zadání aplikace OB ZAVODY

Pro ukázkou vytváření UML diagramů jsem definovala praktický příklad – aplikaci OB ZAVODY. Inspirovala jsem se reálně využívanou webovou aplikací [20], která slouží hlavně pro přihlašování a odhlašování závodníků na závody v orientačním běhu (OB). V současné době je tato aplikace nejčastěji používaným systémem pro přihlašování na závody orientačního běhu v České republice. Její skutečná funkcionality je poměrně široká, proto jsem si pro demonstraci svého návrhu v UML vybrala pouze modul přihlašování a odhlašování závodníků na závody.

Pro účely této práce se můj návrh aplikace OB ZAVODY v některých místech od aplikace využívané v praxi odlišuje.

3.1.1 Popis požadavků na aplikaci OB ZAVODY

Aplikace má poskytovat specifickou funkcionality pro pořadatele závodů, registrované závodníky a neregistrované uživatele. Při vstupu do aplikace uživatel může zvolit jednu z těchto tří možností:

- *přihlásit se jako neregistrovaný uživatel,*
- *přihlásit se jako pořadatel,*
- *přihlásit se jako registrovaný závodník.*

Funkcionality pro neregistrovaného uživatele:

Neregistrovaný uživatel při vstupu do aplikace zvolí možnost *přihlásit se jako neregistrovaný uživatel*.

Aplikace neregistrovanému uživateli nabídne pouze volbu *vypsat všechny závody*. Pod touto volbou se zobrazí kompletní seznam závodů založených v aplikaci, kde u každého závodu jsou k dispozici tyto možnosti:

- *zobrazit informace,*
- *vypsat seznam přihlášených závodníků:*
 - *všech,*
 - *dle oddílu,*
 - *dle kategorie.*

Funkcionality pro pořadatele:

Pořadatel závodu při vstupu do aplikace zvolí možnost *přihlásit se jako pořadatel* a aplikace ho následně vyzve k vyplnění přihlašovacích údajů:

- pořádací oddíl - třípísmenná zkratka oddílu, jedinečná v rámci všech systémů pro závody v OB,
- heslo.

Po úspěšném přihlášení jsou pořadateli k dispozici volby:

- *vypsát všechny závody.*
- *založit nový závod,*
- *vypsát pořádané závody (závody již založené).*

Volba *vypsát všechny závody* pořadateli nabídne stejné možnosti jako neregistrovanému uživateli.

Při *založení nového závodu* aplikace požaduje následující informace (tabulka 3.1):

1. Název závodu	4. Druh závodu – výběr z možností: • jednotlivci, • vícedenní, • štafety.
2. Zařazení do soutěží – výběr z možností: • mistrovské závody, • žebříček E, A, • žebříček B Čechy, • žebříček B Morava, • oblastní závody, • ranking.	5. Rankingový koeficient
	6. Místo konání
	7. Termín řádných přihlášek
	8. E-mail na pořadatele
3. Datum konání	9. Závodní kategorie ² dle platných pravidel OB pro daný druh závodu

Tabulka 3.1 Údaje pro založení nového závodu, zdroj: autor

Po vyplnění těchto údajů a potvrzení je závod založen a je k nalezení pod volbou *vypsát pořádané závody*.

Sekce *vypsát pořádané závody* obsahuje seznam závodů vypsáných pořadatelem. U každého závodu v seznamu je možnost:

- *zobrazit informace,*
- *upravit informace,*
- *zobrazit seznam přihlášených závodníků:*
 - *všech,*
 - *dle oddílu,*
 - *dle kategorie.*
- *upravit seznam přihlášených závodníků:*
 - *přihlásit závodníka,*
 - *odhlásit závodníka,*
 - *změnit přihlašovací údaje závodníka.*

Volbu *upravit seznam přihlášených závodníků* pořadatel využívá:

- a) pokud závodník aplikaci nepoužívá a přihlašuje a odhlašuje se na závody prostřednictvím e-mailu nebo telefonicky,
- b) po vypršení termínu řádných přihlášek, kdy se závodník již prostřednictvím aplikace přihlásit ani odhlásit nemůže, ale má možnost kontaktovat pořadatele, který ho přihlásit i odhlásit může.

Funkcionalita pro registrovaného závodníka:

Registrovaný závodník při vstupu do aplikace zvolí možnost *přihlásit se jako registrovaný závodník* a aplikace ho následně vyzve k vyplnění přihlašovacích údajů:

² Zde nejsou na výběr žádné možnosti, ale předpokládá se, že pořadatel vypíše kategorie dle zvyklostí v dané oblasti.

- registrační číslo - jednoznačná identifikace závodníka v rámci všech systémů pro závody v orientačním běhu,
- heslo.

Po úspěšném přihlášení jsou závodníkovi k dispozici volby:

- *vypsát všechny závody,*
- *vypsát závody, na které je závodník přihlášen.*

Volba *vypsát všechny závody* zobrazí kompletní seznam závodů založených v aplikaci. U každého závodu jsou k dispozici tyto možnosti:

- *zobrazit informace,*
- *vypsát seznam přihlášených závodníků:*
 - *všech,*
 - *dle oddílu,*
 - *dle kategorie.*
- *přihlásit se na závod,*
- *odhlásit se ze závodu,*
- *změnit přihlašovací údaje pro závod.*

a) Pokud závodník na daný závod není zatím přihlášen:

Při volbě *přihlásit se na závod* aplikace požaduje následující informace (tabulka 3.2):

1. kategorie – výběr z kategorií vypsaných pořadatelem
2. jméno – předvyplněno dle registračního čísla zadávaného při přihlašování do aplikace
3. číslo SI ³ + volba z možností: ○ vlastní čip ○ požaduji zapůjčit ○ doplním později
4. licence
5. požadovaný start (nepovinné)
6. poznámka (nepovinné)

Tabulka 3.2 Údaje pro přihlášení závodníka na závod, zdroj: autor

Po vyplnění těchto údajů a potvrzení je závodník přihlášen na závod a tento závod je k nalezení i v sekci *vypsát závody, na které je závodník přihlášen.*

Při volbě *odhlásit se ze závodu* nebo *změnit přihlašovací údaje pro závod*, aplikace oznámí, že se nelze odhlásit ze závodu nebo změnit přihlašovací údaje pro závod, protože závodník zatím není přihlášen na závod.

b) Pokud závodník na daný závod už přihlášen je:

Při volbě *přihlásit se na závod* aplikace oznámí, že závodník již je na závod přihlášen.

Volba *odhlásit se ze závodu* umožní zrušit přihlášku na daný závod a volba *změnit přihlašovací údaje* umožní upravit údaje v přihlášce (např. změnit kategorii).

Závodník se může na závod přihlašovat, odhlášovat a měnit přihlašovací údaje jen do termínu řádných přihlášek. Po termínu řádných přihlášek se už závodník na závod prostřednictvím aplikace přihlásit ani odhlásit nemůže, ale má možnost kontaktovat pořadatele, který ho přihlásit i odhlásit může.

³Číslo čipu (jedinečné), s nímž závodník poběží.

Volba vypsat závody, na které je závodník přihlášen zobrazí seznam závodů, na které je závodník momentálně přihlášen. U každého závodu z tohoto seznamu jsou na výběr stejné možnosti jako v sekci vypsat všechny závody.

3.1.2 Slovníček pojmů

Každá řešená oblast má vlastní terminologii a při řešení daného projektu je třeba ji sjednotit. Klíčové termíny a jejich definice je proto vhodné zachytit do slovníčku pojmů (project glossary), kterému by měli rozumět všichni účastníci projektu, včetně uživatelů. Je důležité zajistit, aby ve slovníčku definované pojmy byly synchronizovány s UML modelem a vůbec v rámci celého projektu. [1]

Pojem	Vysvětlení	Alternativní pojem/zkratka
Orientační běh	„Orientační běh je sport, jehož podstatou je spojení běhu s orientací v neznámém terénu. Závodníci při něm za pomoci mapy a buzoly absolvují trať v terénu (převážně zalesněném) určenou startem, kontrolami a cílem. Závodník se během svého výkonu pohybuje libovolně terénem, omezen může být povinnými úseky a zakázanými prostory. Cílem výkonu je absolvování závodní trati v minimálním čase.“ [21]	OB
(OB) Závod	Sportovní akce v orientačním běhu.	
Mistrovský závod	Mistrovství České republiky v určité disciplíně orientačního běhu (např. krátká trať, klasika, štafety).	MČR
Žebříček E/ A/ B Čechy/ B Morava	Dlouhodobé republikové soutěže pro držitele licencí E/R/A/B.	ŽE, ŽA, ŽB Čechy, ŽB Morava
Oblastní závod	Závod v rámci oddílů spadajících do jedné oblasti. Jeho výsledky jsou zařazeny do oblastního žebříčku.	
Ranking	Dlouhodobá soutěž jednotlivců. Absolutní pořadí mužů a žen dospělých kategorií v České republice ze všech závodů dle určitých kritérií za 1 rok.	
Závod jednotlivců	Soutěž jednotlivců, kteří závodí každý sám za sebe.	
Vícedenní závod	Závod na několik etap, konaný ve dvou a více dnech.	
Štafety	Soutěž dvou a více jednotlivců, kteří tvoří tým a startují v závodě po sobě.	
Závodník	Osoba, která je registrovaná v Českém svazu orientačních sportů, má tedy registrační číslo a může se přihlásit do aplikace jako registrovaný závodník.	Běžec
Pořadatel	Osoba(y), která(é) se podílí na přípravě a průběhu závodů.[21] Identifikací pořadatele je třípísmenná zkratka, jedinečná ve všech systémech pro OB, a skrze ní se může také přihlásit do aplikace jako pořadatel.	
Oddíl	Sdružení osob zainteresovaných v orientačních sportech, které má svého předsedu a vystupuje jako celek. Nepočítaje velkoměsta obvykle platí „co oddíl, to jiné město“.	Klub
Rankingový koeficient	Jedno z kritérií pro ranking. Obecně platí, že čím vyšší, tím je závod náročnější.	
(závodní) Kategorie	Rozřazení závodníků podle pohlaví a věku.	

Registrační číslo	Jednoznačná identifikace závodníka v rámci všech systémů pro orientační běh.	
(jednotka) SI	Elektronický čip, který má závodník při závodě na prstě. Na každé kontrole jím označuje svůj průchod.	Čip
Licence	Výkonnostní třída závodníka. V pořadí od nejlepších: E, R, A, B, C	

Tabulka 3.3 Slovníček pojmů, zdroj: autor

3.2 Diagram případů užití

Diagram případů užití (Use Case Diagram) zachycuje chování systému z hlediska uživatele. Prvky diagramu případů užití jsou aktéři (Actors), případy užití (Use Cases) a vztahy (Relations). Aktér představuje roli, ve které vystupuje jeden typ uživatele v rámci své komunikace se systémem. Příklad užití reprezentuje určitou funkčnost, která má být systémem pokryta. Všechny případy užití umožňují znázornit funkční požadavky na systém. [3], [6]

V rámci analýzy se vytváří model případů užití, který se skládá nejen z diagramů případů užití, ale i ze specifikací (slovních popisů) jednotlivých případů užití. [3], [25]

Podrobnější charakteristika diagramu případů užití a jeho prvků je přehledně zpracována např. v [3].

3.2.1 Model případů užití aplikace OB ZAVODY

V diagramu případů užití aplikace OB ZAVODY, který je zachycen na obrázku 3.1, jsou čtyři aktéři:

- *Uživatel*,
- *Registrovaný uživatel*,
- *Závodník*,
- *Pořadatel*.

Mezi aktéry je použit vztah **generalizace/specializace**.

Aktér *Registrovaný uživatel* je zobecněním aktérů *Závodník* a *Pořadatel*. Jedná se o abstraktního aktéra, jehož použití v diagramu umožní zakreslit jen jednu komunikační vazbu mezi aktérem *Registrovaný uživatel* a případem užití *Přihlásit se k aplikaci*.

Aktér *Uživatel* je zobecněním aktéra *Registrovaný uživatel*. Případy užití *Vypsát všechny závody*, *Zobrazit detailní informace závodu* a *Zobrazit seznam přihlášených na závod*, se kterými aktér *Uživatel* komunikuje, jsou společné pro všechny uživatele aplikace, včetně neregistrovaných uživatelů. Nejedná se tedy o aktéra abstraktního, protože jeho roli v aplikaci vykonává reálně existující neregistrovaný uživatel.

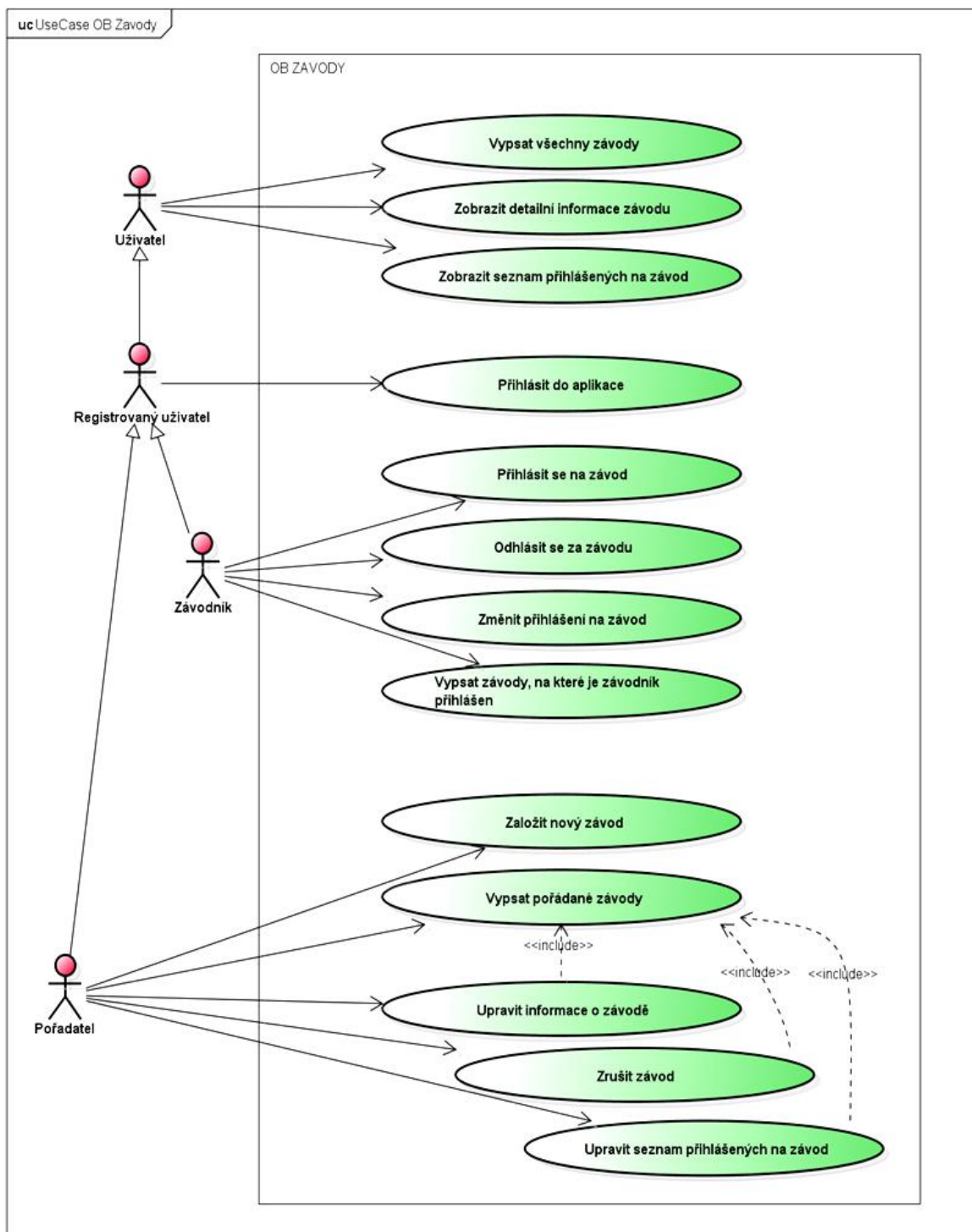
Diagram případů užití aplikace OB ZAVODY obsahuje tyto případy užití (tabulka 3.4):

UC1	<i>Vypsát všechny závody</i>
UC2	<i>Zobrazit detailní informace závodu</i>
UC3	<i>Zobrazit seznam přihlášených na závod</i>
UC4	<i>Přihlásit do aplikace</i>
UC5	<i>Přihlásit se na závod</i>
UC6	<i>Odhlásit se ze závodu</i>
UC7	<i>Změnit přihlášení na závod</i>
UC8	<i>Vypsát závody, na které je závodník přihlášen</i>
UC9	<i>Založit nový závod</i>
UC10	<i>Vypsát pořádané závody</i>

UC11	<i>Upravit informace o závodě</i>
UC12	<i>Zrušit závod</i>
UC13	<i>Upravit seznam přihlášených na závod</i>

Tabulka 3.4 Případy užití, zdroj: autor

Případy užití UC11 *Upravit informace o závodě*, UC12 *Zrušit závod* a UC13 *Upravit seznam přihlášených na závod* jsou propojené vazbou typu <<include>> s případem užití UC10 *Vypsát pořádané závody*. Relace << include>> je v diagramu z toho důvodu, že pořadatel má právo upravovat informace o závodě, zrušit závod a upravovat seznam přihlášených jen u těch závodů, které pořádá, nikoli u všech závodů.

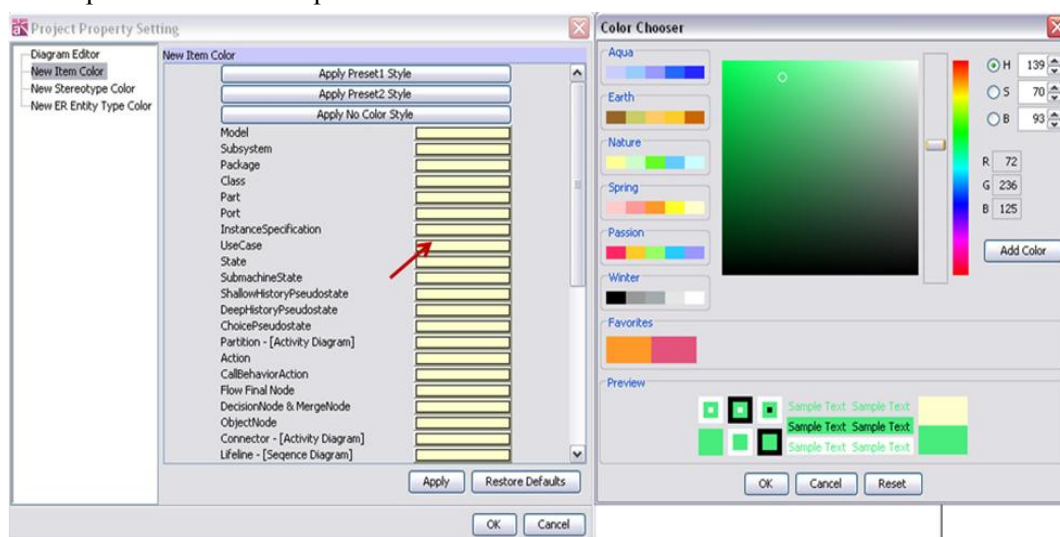


Obrázek 3.1 Diagram případů užití, zdroj: autor

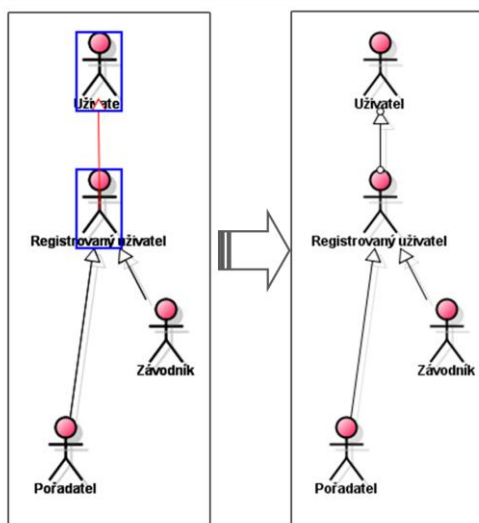
3.2.2 Postup tvorby diagramu případů užití v Astah* professional

Diagram případů užití aplikace OB ZAVODY, zachycený na obrázku 3.1, můžete v programu Astah* professional vytvořit v následujících krocích:

1. Založte nový projekt a uložte ho pod názvem např. OBzavody, viz kapitola 2.2.3.
2. V hlavní nabídce vyberte [Diagram] – [UseCase Diagram].
V editoru diagramu se zobrazí okno *UseCase Diagram 0/ UseCase Diagram* pro návrh diagramu případů užití, dále jen diagram případů užití.
3. Pokud chcete jednotlivé typy prvků diagramu (v tomto případě aktéry a případy užití) na první pohled odlišit a diagram tak oživit, můžete nastavit barvu pro aktéry a případy užití v tomto projektu:
 - v hlavní nabídce zvolte [Tool] – [Project] – [Set Project Properties], což vyvolá dialogové okno *Project Property Setting*, (viz obrázek 3.2),
 - v dialogovém okně *Project Property Setting* vyberte položku [New Item Color] – [UseCase] a v nově zobrazeném okně *Color Choser* zvolte pro případy užití např. světle zelenou barvu (viz obrázek 3.2),
 - v dialogovém okně *Project Property Setting* dále vyberte položku [New Stereotype Color] – [actor] a v okně *Color Choser* zvolte (obdobně jako v předchozím kroku) např. červenou barvu pro aktéra.

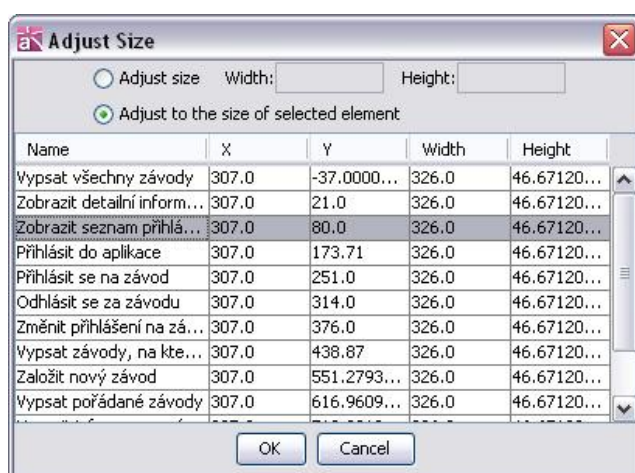


Obrázek 3.2 Nastavení barvy případu užití, zdroj: prostředí programu Astah* professional



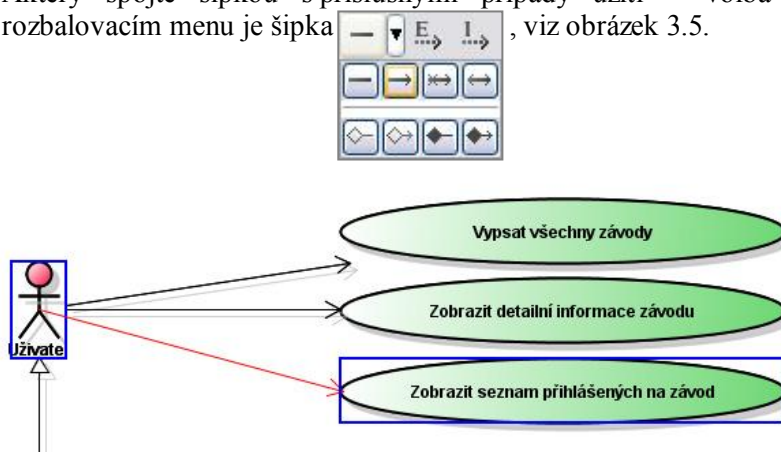
Obrázek 3.3 Generalizace/specializace aktérů, zdroj: prostředí programu Astah* professional

4. Do diagramu případů užití vložte čtyři aktéry – volba panáčka , pojmenujte je a rozmístěte pod sebe. Následně aktéry spojte vztahem generalizace/specializace – volba prázdné šipky nahoru  jako na obrázku 3.3.
5. Do diagramu případů užití vložte 13 případů užití – volba elipsy , pojmenujte je a umístěte pod sebe napravo od aktérů.
6. Z estetických důvodů všem případům užití doporučuji nastavit stejnou velikost:
 - označte všechny případy užití,
 - v hlavní nabídce zvolte [Alignment] – [Adjust Size] – [Adjust Size]
 - ve vyvolaném dialogovém okně s názvem *Adjust Size* zvolte možnost přizpůsobení velikosti prvků diagramu podle jednoho vybraného prvku – [Adjust to the size of the selected element] a z nabídnuté tabulky vyberte např. případ užití *Zobrazit seznam přihlášených na závod* (viz obrázek 3.4).



Obrázek 3.4 Přizpůsobení velikosti, zdroj: prostředí programu Astah* professional

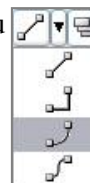
7. Aktéry spojte šipkou s příslušnými případy užití – volba vodorovné čáry, v jejímž rozbalovacím menu je šipka , viz obrázek 3.5.

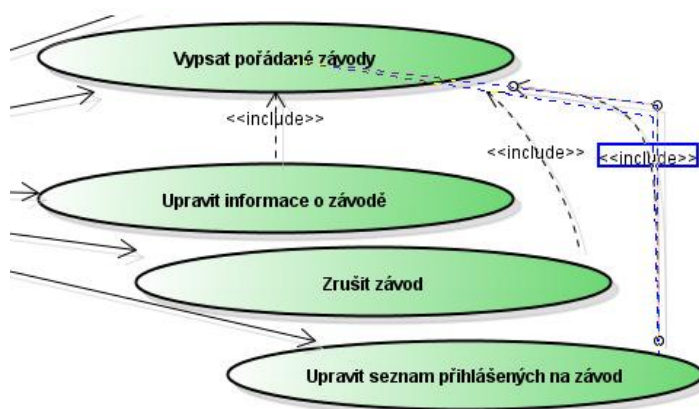


Obrázek 3.5 aktérů s případy užití, zdroj: prostředí programu Astah* professional

8. Poslední tři případy užití *Upravit informace o závodě*, *Zrušit závod* a *Upravit seznam přihlášených na závod* spojte vztahem include – volba I s přerušovanou šipkou 

s případem užití *Vypsat pořádané závody*. Zakřivený tvar šipky naleznete pod volbou šikmé čáry (vpravo v nabídce), kde z rozbalovacího menu vyberte křivku





Obrázek 3.6 Vztah include se zakřivenou šipkou, zdroj: prostředí programu Astah* professional

9. Případy užití doporučuji zarovnat, aby byly pod sebou:
 - všechny případy užití, kromě posledních dvou se vztahem include, označte,
 - v hlavní nabídce zvolte [Alignment] – [Align Vertically] – [Align Left].
10. Všechny případy užití orámujte obdélníkem, který značí hranici systému – volba obdélníku  a do jeho levého horního rohu umístíte text OB ZAVODY – volba , viz finální obrázek 3.1.
11. Hotový diagram případů užití můžete vyexportovat do formátu JPEG, viz kapitola 2.2.7.

3.2.3 Ukázková specifikace případu užití Přihlásit se na závod

Součástí modelu případů užití jsou rovněž specifikace jednotlivých případů užití (scénáře případů užití), které představují sekvenci kroků popisujících interakci mezi aktérem a systémem.[6]

Jazyk UML nedefinuje žádný standardní způsob, jak specifikaci případu užití zapisovat a v různých případech jsou vhodné různé formáty. [6], [4]

V tabulce 3.5 je zachycena ukázková specifikace případu užití *UC5 Přihlásit se na závod*, jejíž formát je převzatý z [3].

Název případu užití	Přihlásit se na závod		
Identifikace případu užití	UC5		
Cíl případu užití	Přihlásit se na vybraný závod		
Primární aktér(ři)	Závodník		
Pomocný aktér(ři)	-		
Vstupní podmínky	Závodník je přihlášen k aplikaci, zvolí <i>vypsát všechny závody</i> a u vybraného závodu ze seznamu zadá volbu <i>přihlásit se na závod</i>		
Výstupní podmínky	Závodník je přihlášen na vybraný závod		
Základní scénář	Krok	Role	Akce
	1	System	kontroluje, zda již nevypršel termín řádných přihlášek
	2	System	pokud termín řádných přihlášek nevypršel, kontroluje, zda závodník není na tento závod již přihlášen
	3	System	pokud závodník není na tento závod přihlášen,

			zobrazí formulář pro přihlášení na závod, kde jméno závodníka je předvyplněno dle registračního čísla, pod kterým se přihlašuje do aplikace
	4	Aktér	zadá údaje: kategorie, číslo SI, jednu z možností vlastní čip/požadují zapůjčit/doplňím později, licence, požadovaný start (nepovinně), poznámka (nepovinně)
	5	Aktér	zvolí, že chce dokončit přihlášku
	6	System	provede kontrolu zadaných údajů
	7	System	pokud není chyba, uloží přihlášku do systému
Alternativní scénář 2a: Termín řádných přihlášek vypršel	Krok	Role	Akce
	2a1	System	pokud vypršel termín řádných přihlášek, zobrazí hlášení „Prostřednictvím aplikace se již nelze přihlásit, protože termín řádných přihlášek vypršel. Pro dodatečné přihlášení kontaktujte pořadatele.“
	2a2	System	nabídne kontakt na pořadatele
Alternativní scénář 3b: Závodník je již přihlášen na závod	Krok	Role	Akce
	3b1	System	pokud je závodník již na závod přihlášen, zobrazí hlášení „Závodník <i>jméno příjmení</i> je již na závod přihlášen.“
	3b2	System	nabídne volbu <i>odhlásit se závodu</i> (viz UC6 Odhlásit se závodu) a volbu <i>změnit přihlašovací údaje závodníka</i> (viz UC7 Změnit přihlášení na závod)
Alternativní scénář 7c: Chyba vstupních údajů	Krok	Role	Akce
	7c1	System	pokud je chyba vstupních údajů, zobrazí chybové hlášení
	7c2	Aktér	opraví položku(y), k níž (nimž) se vztahuje chybové hlášení
	7c3	Aktér	pokračuje od bodu 5 základního scénáře

Tabulka 3.5 Specifikace případu užití, zdroj: autor dle šablony [3]

Možnosti uchovávání specifikací případů užití:

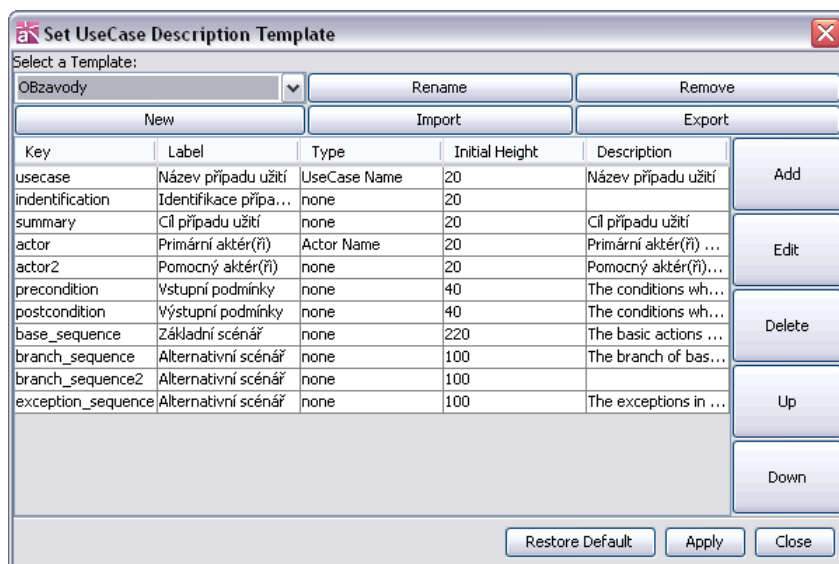
- Specifikace případů užití je možné vytvářet a uchovávat externě, bez propojení s případy užití. Takto vytvořený model případů užití s jejich specifikacemi můžete udržovat např. pomocí adresářové struktury operačního systému.
- Specifikace případů užití můžete vytvořit externě, a následně je importovat a připojit k jednotlivým případům užití v CASE nástroji, v němž jsou potom uchovávány s nimi propojené.
- Specifikace případů užití vytvořit přímo v CASE nástroji, kde je můžete snadno uchovávat propojené s jednotlivými případy užití.

Následuje postup vytvoření specifikace případů užití v Astah* professional:

- v lokálním menu případu užití zvolte [*Open UseCase Description*],
- otevře se okno s názvem *NázevPřípaduUziti/ UseCase Description*, ve kterém vyplníte jednotlivé položky specifikace případu užití.

Program umožňuje i vytváření vlastních šablon pro specifikace případů užití:

- v hlavní nabídce zvolte [*Tool*] – [*Set Template*] – [*UseCase Description*]
- otevře se okno s názvem *Set UseCase Description Template* jako na obrázku 3.7, ve kterém lze nadefinovat vlastní specifikace případů užití.



Obrázek 3.7 Nastavení šablony specifikace případů užití, zdroj: prostředí programu Astah* professional

Na ukázkou jsem v programu nadefinovala šablonu specifikace případu užití *UC5 Přihlásit se na závod* stejnou jako v tabulce 3.5. Zachycuje ji obrázek 3.8.

ITEM	VALUE
Název případu užítí	Přihlásit se na závod
Identifikace případu užítí	UC5
Cíl případu užítí	Přihlásit se na vybraný závod
Primární aktér(ři)	Závodník
Pomocný aktér(ři)	-
Vstupní podmínky	Závodník je přihlášen k aplikaci, zvolí vypsat všechny závody a u vybraného závodu se seznamu zadá volbu přihlásit se na závod
Výstupní podmínky	Závodník je přihlášen na vybraný závod
Základní scénář	Krok 1 - Systém: kontroluje, zda již nevypršel termín řádných přihlášek Krok 2 - Systém: pokud termín řádných přihlášek nevypršel, kontroluje, zda závodník není na tento závod již přihlášen Krok 3 - Systém: pokud závodník není na tento závod přihlášen, zobrazí formulář pro přihlášení na závod, kde jméno závodníka je předvyplněno dle registračního čísla, pod kterým se přihlašuje do aplikace Krok 4 - Aktér: zadá údaje: kategorie, číslo SI, jednu z možností vlastní čip/pořadují zapůjčit/doplňím později, licence, požadovaný start (nepovinně), poznámka (nepovinně) Krok 5 - Aktér: zvolí, že chce dokončit přihlášku Krok 6 - Systém: provede kontrolu zadaných údajů Krok 7 - Systém: pokud není chyba, uloží přihlášku do systému
Alternativní scénář	1a - Termín řádných přihlášek vypršel Krok 1a1 - Systém: pokud vypršel termín řádných přihlášek, zobrazí hlášení „Prostřednictvím aplikace se již nelze přihlásit, protože termín řádných přihlášek vypršel. Pro dodatečné přihlášení kontaktujte pořadatele.“ Krok 1a2 - Systém: nabídne kontakt na pořadatele
Alternativní scénář	3b - Závodník je již přihlášen na závod Krok 3b1 - Systém: pokud je závodník již na závod přihlášen, zobrazí hlášení „Závodník jméno příjmení je již na závod přihlášen.“ Krok 3b2 - Systém: nabídne volbu odhlásit se závodu (viz UC6 Odhlásit se závodu) a volbu změnit přihlašovací údaje závodníka (viz UC7 Změnit přihlášení na závod)
Alternativní scénář	7c - Chyba vstupních údajů Krok 7c1 - Systém: pokud je chyba vstupních údajů, zobrazí chybové hlášení Krok 7c2 - Aktér: opraví položku(y), k níž (nimž) se vztahuje chybové hlášení Krok 7c3 - Aktér: pokračuje od bodu 5 základního scénáře

Obrázek 3.8 Specifikace případu užítí v Astah* professional, zdroj: autor

3.3 Diagram tříd

Diagram tříd (Class Diagram) představuje „statický pohled na modelovaný systém“.[3] Zobrazuje strukturu objektových tříd a jejich vzájemné vztahy v systému.[6]

Třída je definována svými atributy, operacemi a vztahy k ostatním třídám, které jsou společné pro určitou množinu objektů stejného typu. Lze ji chápat jako předpis, podle kterého se vytváří konkrétní objekt (instance). [3], [6]

Podle [3] se dle úrovně abstrakce rozlišují tři úrovně modelu tříd:

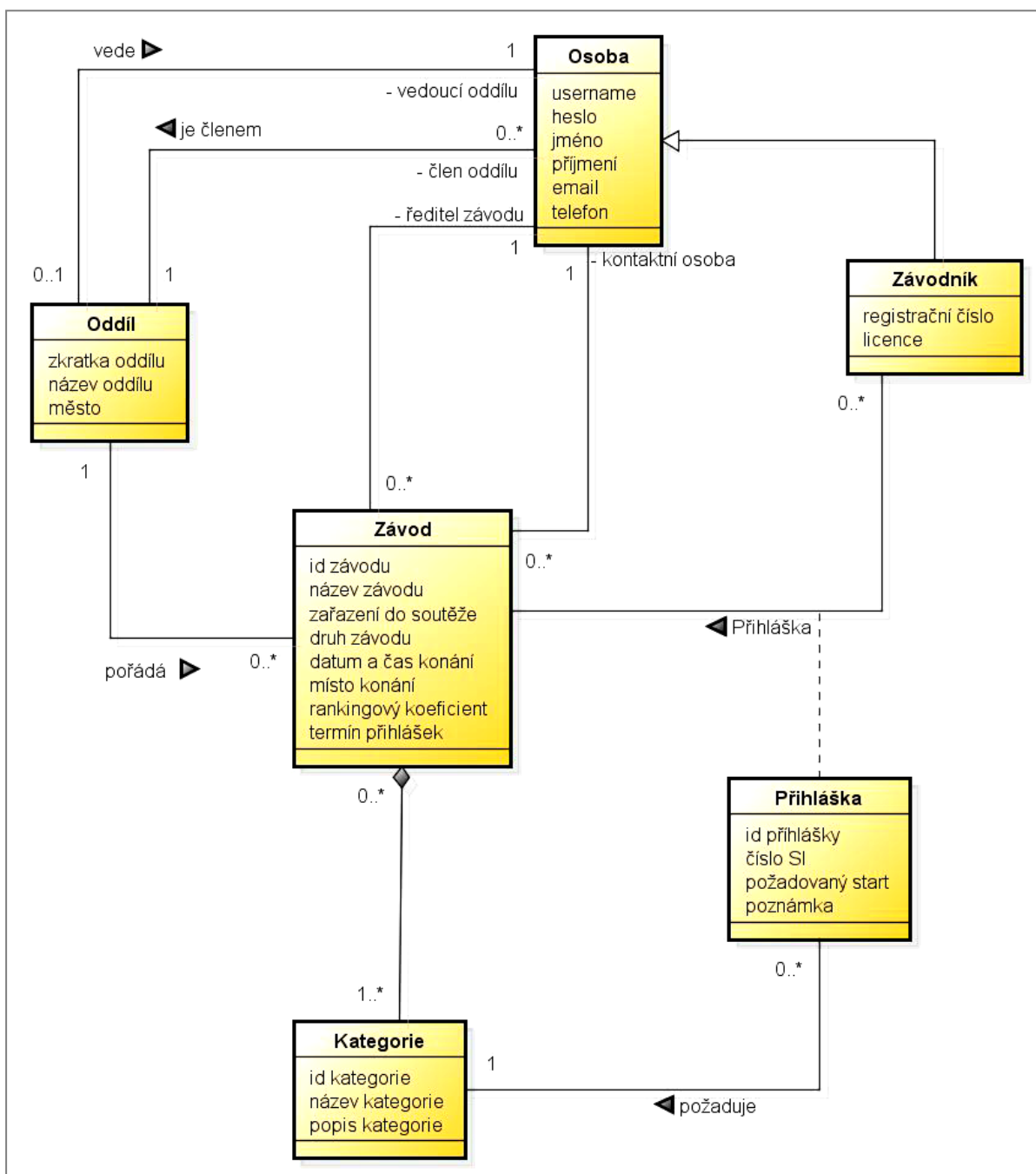
- konceptuální,
- designová,
- implementační.

Konceptuální model tříd (analytický, doménový) má za úkol zmapovat třídy zastupující objekty z problémové oblasti bez implementačních detailů. Jde tedy o model zcela implementačně nezávislý.

U **designového modelu tříd** se původní analytické třídy a jejich vazby zpřesňují a rozšiřují. Často jsou také přidávány třídy další. Designový model již má některé implementační charakteristiky.

Implementační model tříd obsahuje veškeré implementační charakteristiky pro případné vygenerování zdrojového kódu ve vybraném programovacím jazyce.

Podrobnější charakteristika diagramu tříd a jeho prvků je přehledně zpracována např. v [3].



Obrázek 3.9 Analytický model tříd, zdroj: autor

3.3.1 Model tříd aplikace OB ZAVODY

Na obrázku 3.9 je zachycen konceptuální model tříd aplikace OB ZAVODY.

Model tříd zahrnuje celkem 6 byznys tříd, mapujících problémovou oblast:

- *Osoba*,
- *Závodník*,
- *Závod*,
- *Oddíl*,
- *Přihláška*,
- *Kategorie*.

Strukturu tříd tvoří pouze atributy, které určují odpovědnosti tříd. Protože konceptuální model je implementačně nezávislý, jsou zde definovány jenom názvy atributů. Nejsou určeny viditelnosti atributů, ani jejich datové typy a počáteční hodnoty.

Vztahy mezi třídami jsou pojmenovány a jsou u nich určeny jejich násobnosti.

Mezi třídami *Osoba* a *Závodník* je použit vztah **generalizace/ specializace**. Členy oddílu potom mohou být kromě závodníků i osoby, které se nezúčastňují závodů, ale podílí se např. na pořádání závodů.

Třídy *Závod* a *Kategorie* jsou propojeny vztahem **kompozice**. Jedná se o případ, že pro každý závod se vypisují kategorie (dle uvážení pořadatele), které si daný závod drží. *Kategorie* tedy nemůže existovat samostatně bez třídy *Závod*, ale vždy jen jako jeho součást.

V modelu je použita i jedna **asociační třída** *Přihláška*, která obsahuje informace vztahující se k přihlášení závodníka na závod.

Třídy *Oddíl* a *Osoba* jsou propojeny dvěma vztahy **asociace**. První z nich znázorňuje, že *Oddíl* vede *Osoba*. Druhý vztah znamená, že *Osoba* je členem oddílu.

Všechny ostatní vztahy jsou zachyceny rovněž jako **asociace**.

3.3.2 Postup tvorby diagramu tříd v Astah* professional

Diagramu tříd aplikace OB ZAVODY, zachycený na obrázku 3.9, můžete v programu Astah* professional vytvořit v následujících krocích:

1. V projektu OBzavody, který jste vytvořili pro diagram případů užití (viz kapitola 3.2.2) založte nový balíček⁴ s názvem *Class Diagram*. Postup založení balíčku v projektu je popsán v kapitole 2.2.3.
2. Do balíčku *Class Diagram* vložte diagram tříd (viz kapitola 2.2.3).
V editoru diagramu se zobrazí okno *Class Diagram 0/ Class Diagram* pro návrh diagramu tříd, dále jen diagram tříd.
3. V konceptuálním modelu se mají definovat pouze názvy atributů a operací, ale v programu je výchozí zobrazení atributů a operací ve třídě přednastaveno ve tvaru:

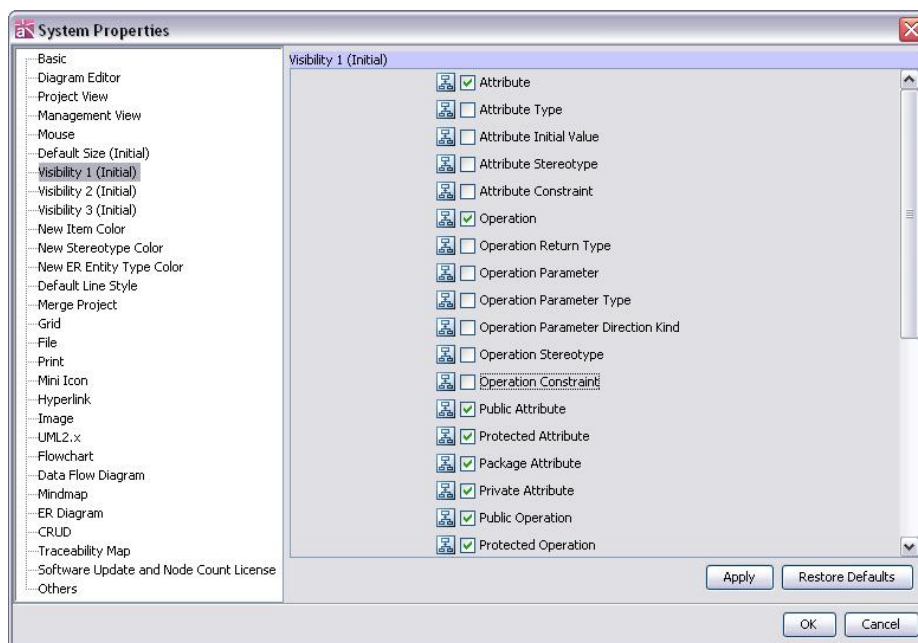
Atributy: *viditelnost* *název*: *typ*

Operace: *viditelnost* *název*: *návratový typ*

Je proto vhodné nastavení upravit tak, aby byl ve třídě zobrazen pouze název atributu nebo metody:

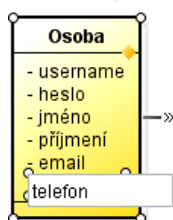
⁴ Nový balíček je vhodné založit z toho důvodu, že prvky modelu musí mít jedinečný název v rámci jednoho jmenného prostoru. Pokud nezaložíte nový balíček pro diagram tříd, nemůžete mít v diagramu tříd třídu *Závodník*, protože její název se shoduje s názvem aktéra v diagramu případů užití.

- v hlavní nabídce zvolte [Tool] – [System Properties], což vyvolá dialogové okno pro nastavení vlastností systému s názvem *System Properties*, viz obrázek 3.10,
- v okně *System Properties* vyberte položku [Visibility 1 (Initial)]
- odškrtněte možnosti [Attribute Type], [Attribute Initial Value], [Attribute Stereotype], [Attribute Constraint], [Operation Return Type], [Operation Parameter], [Operation Parameter Type], [Operation Stereotype], [Operation Constraint], viz obrázek 3.10



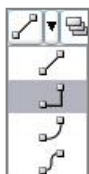
Obrázek 3.10 Vlastnosti systému – viditelnost atributů a operací, zdroj: prostředí programu Astah* professional

4. Pokud nejste spokojeni s výchozí barvou pro třídy, můžete ji změnit (např. na oranžovou). Postup je stejný jako u nastavení barvy pro případy užití v kapitole 3.2.2.
5. Do diagramu tříd vložte 5 tříd – volba , a pojmenujte je *Osoba*, *Závodník*, *Oddíl*, *Závod* a *Kategorie*.
6. Do jednotlivých tříd vepište jejich atributy. Při najetí myši na prvek třídy se uvnitř třídy objeví oranžový kosočtverec a pod ním zelený obdélník. Kliknutím na kosočtverec se v horní sekci třídy vytvoří atribut, viz obrázek 3.11. (Operace se vytvoří v dolní sekci kliknutím na zelený obdélník.)

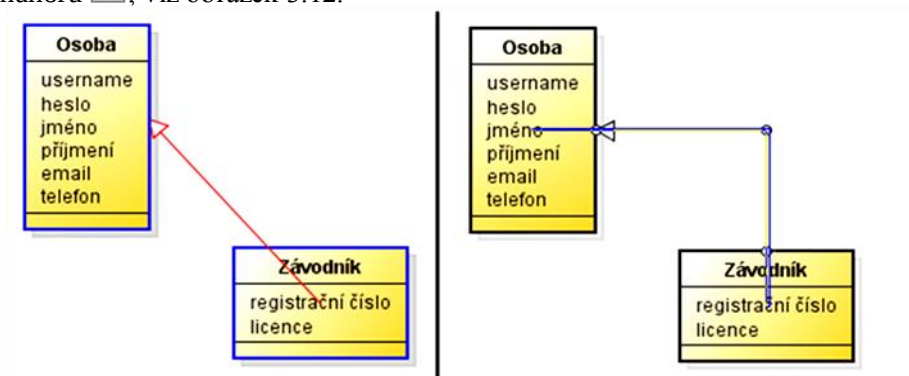


Obrázek 3.11 Vepsání atributů do třídy, zdroj: prostředí programu Astah* professional

7. U atributů ve třídách se zobrazuje pomlčka, což je symbol pro viditelnost *Private*. Zobrazení této pomlčky je možné vypnout pouze v lokální nabídce třídy – volba [Extended Visibility], kde se odznačí [Visibility Kind Visibility].
8. Pro modelování vztahů mezi třídami je dobré pro lepší přehlednost digramu použít pravoúhlý typ čar. V pravé části menu je volba šikmé čáry, v jejímž rozbalovacím menu vyberete pravoúhlý typ čáry .



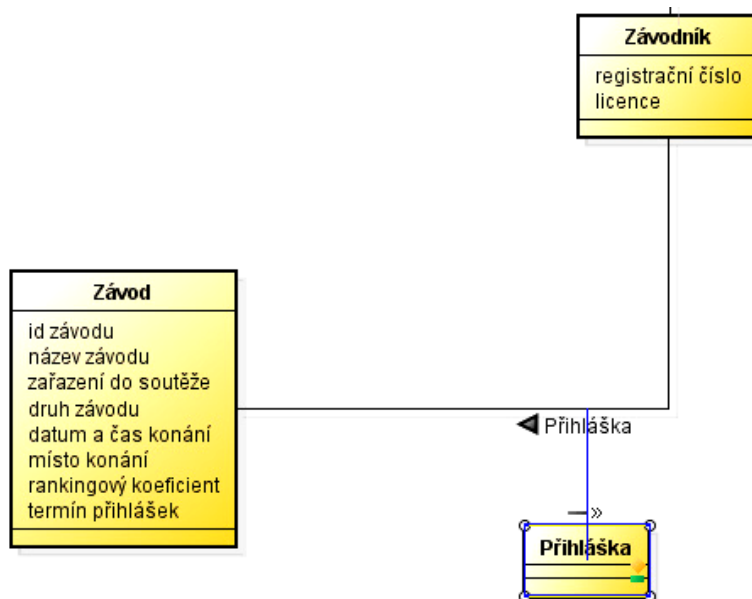
9. Třidu *Osoba* a *Závodník* spojte vztahem generalizace/specializace – volba prázdné šipky nahoru , viz obrázek 3.12.



Obrázek 3.12 Generalizace/specializace mezi třídami, zdroj: prostředí programu Astah* professional

10. Třidu *Závodník* a *Závod* spojte vazbou s asociační třídou – volba .

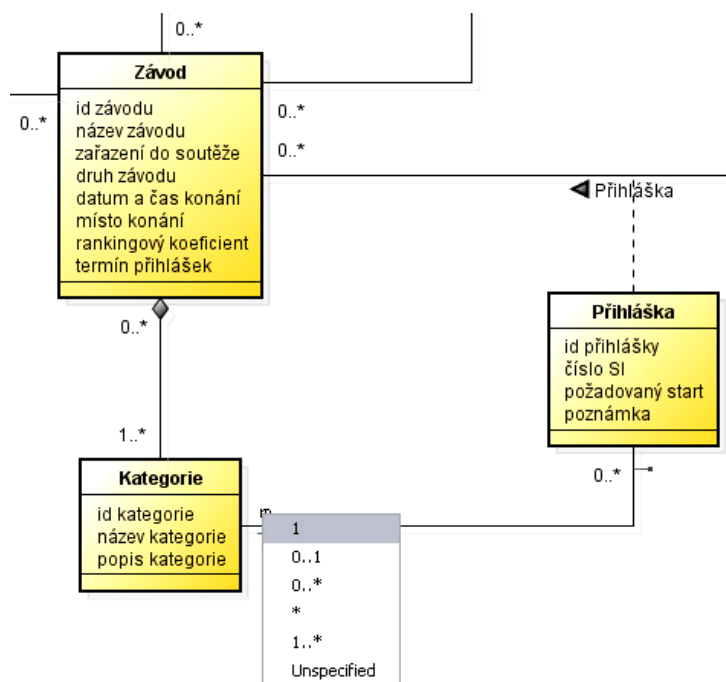
Všechny vazby v diagramu tříd vždy vytvářejte ve směru jejich čtení (v tomto případě směrem od *Závodníka* k *Závodu*), protože po jejich vytvoření již směr vazby nelze změnit. Asociační třídu pojmenujte *Přihláška*. Automaticky se pojmenuje i vazba. Šipka v názvu vazby směřuje od *Závodníka* k *Závodu*, viz obrázek 3.13.



Obrázek 3.13 Asociační třída Přihláška, zdroj: prostředí programu Astah* professional

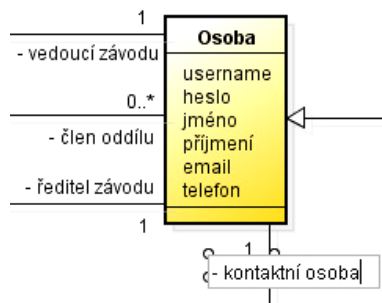
11. Do asociační třídy *Přihláška* vepište atributy, stejným způsobem jako u ostatních tříd.
12. Vazbou asociace – volba vodorovné čáry  spojte třídy:
- *Přihláška* a *Kategorie* (ve směru od *Přihlášky* ke *Kategorii*),
 - *Oddíl* a *Závod* (ve směru od *Oddílu* k *Závodu*).
13. Dvěma vazbami asociace spojte třídy:
- *Závod* a *Osoba* (obě ve směru od *Závodu* k *Osobě*),

- *Oddíl* a *Osoba* (první ve směru od *Oddílu* k *Osobě*, druhá ve směru od *Osoby* k *Oddílu*).
14. Třídou *Závod* a *Kategorie* propojte vazbou kompozice (ve směru od *Závodu* ke *Kategorii*) – volba vodorovné čáry, v jejímž rozbalovacím menu vyberete čáru začínající plným kosočtvercem
15. U všech vazeb určete jejich násobnosti. Po označení dané vazby se na jejích koncích objeví malá písmenka *m*. Po kliknutí na písmenko *m* se zobrazí nabídka násobností, ze které vyberete požadovanou násobnost, viz obrázek 3.14.



Obrázek 3.14 Určení násobnosti vazeb, zdroj: prostředí programu Astah* professional

16. Do třídy *Osoba* vstupují celkem čtyři vazby. Proto je dobré pojmenovat jednotlivé role, které *Osoba* ve vazbách zastupuje. Po označení dané vazby se na jejích koncích, kromě písmenka *m*, objeví i „špendlík“. Klikněte na „špendlík“ a na konec vazby připište název role, viz obrázek 3.15



Obrázek 3.15 Určení role třídy ve vazbě, zdroj : prostředí programu Astah* professional

17. Pro lepší pochopení účelu vazby doplňte k vazbám jejich popisy⁵:

⁵ Upozorňuji, že při tvorbě popisů vazeb program není příliš přívětivý.

- vazbu označte,
- z nabídky vyberte volbu písmene L ,
- několikrát klikněte na danou vazbu, dokud se celý diagram tříd jakoby „nerozostří“,
- napište název vazby a potvrďte enterem,
- teprve po potvrzení enterem se název vazby v diagramu objeví a můžete ho libovolně editovat.

Pozn.: Směr šipky u názvu vazby editovat nelze. Řídí se směrem, kterým byla vazba vytvořena.

18. Třídy a vazby uspořádejte tak, aby byl diagram přehledný, viz finální obrázek 3.9 a můžete ho vyexportovat do formátu JPEG, viz kapitola 2.2.7.

3.4 Diagram aktivit

Diagram aktivit (Activity Diagram) zobrazuje posloupnost aktivit, které mohou probíhat jak sekvenčně, tak paralelně.[3] Využívá se při modelování byznys procesů a pracovních postupů, pro popis procedurální logiky, či pro modelování logiky scénářů případů užití. [25], [3] Jedná se v podstatě o „objektově orientovaný vývojový diagram“. [1]

Prvky, se kterými se setkáte v diagramu aktivit, jsem zapsala do tabulky 3.6:

Název prvku	Alternativní název (/názvy) prvku	Název prvku v angličtině
Zahájení a ukončení	Počáteční a koncový uzel	Initial node and activity final node
Aktivita	Akce, Akční uzel	Activity
Tok	Přechod, Hrana	Flow
Rozhodnutí a sloučení	Hodnocení přechodů, Rozhodování, Uzel rozhodnutí a sloučení	Decision and merge
Větvení a spojení	Uzel rozvětvení a spojení	Fork and join
Plavecké dráhy	Oddíly aktivit	Swimlanes, Partition
Objekt (méně časté)	Objektový uzel	Object

Tabulka 3.6 Prvky diagramu aktivit, zdroj: [1], [3], [6]

Podrobnou charakteristiku diagramu aktivit a jeho prvků naleznete např. v [1].

3.4.1 Diagram aktivit aplikace OB ZAVODY

Na obrázku 3.16 je zachycen diagram aktivit s plaveckými drahami, který popisuje hlavní byznys proces aplikace.

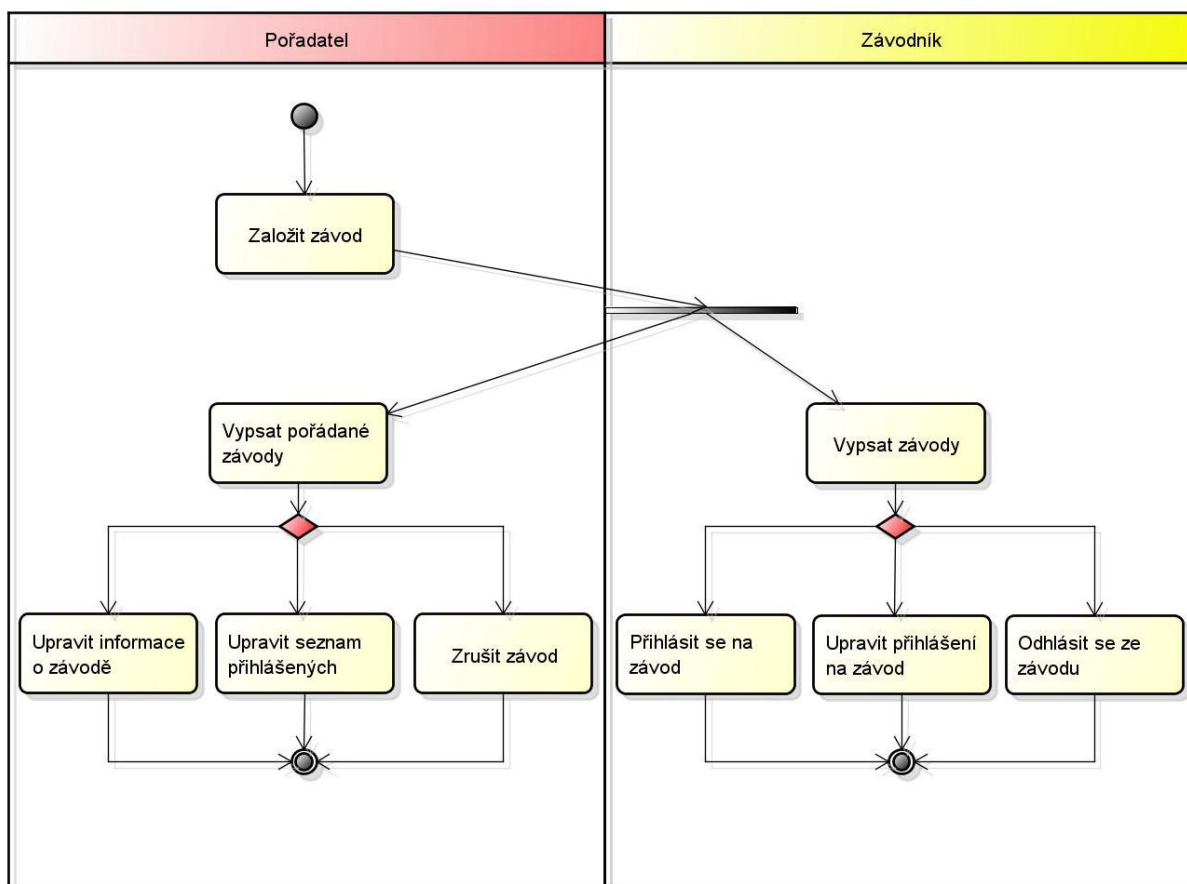
V levém pruhu (plavecké dráze) jsou aktivity, které provádí pořadatel a v pravém pruhu aktivity, které provádí závodník.

Byznys proces je zahájen akcí *Založit závod*, kterou vykonává pořadatel.

Po dokončení akce *Založit závod* dochází k rozvětvení hlavní cesty do dvou souběžných a navzájem nezávislých toků. Závodníkem jsou *Vypisovány závody* a pořadatelem jsou *Vypisovány pořádané závody*.

Po *Vypsání pořádaných závodů* má pořadatel možnost *Upravit informace o závodě*, nebo *Upravit seznam přihlášených*, nebo *Zrušit závod*. Vždy si ovšem musí vybrat jen jednu z variant.

Po *Vypsání závodů* má závodník možnost *Přihlásit se na závod*, nebo *Upravit přihlášení na závod*, nebo *Odhlásit se ze závodu*. Stejně jako pořadatel, i závodník má na výběr vždy jen jednu z možností.



Obrázek 3.16 aktivit hlavního byznys procesu aplikace OB ZAVODY, zdroj: autor dle [3]

Obrázek 3.17 zachycuje diagram aktivit pro případ užití UC5 *Přihlásit se na závod*, vycházející ze specifikace případu užití, vytvořené v kapitole 3.2.1.

Plavecké dráhy jsem v tomto případě vynechala. Bez nich je model v tomto případě přehlednější a odpovědnosti za aktivity jsou zahrnuty již ve specifikaci případu užití.

Případ užití je zahájen akcí *Přihlásit se na závod*.

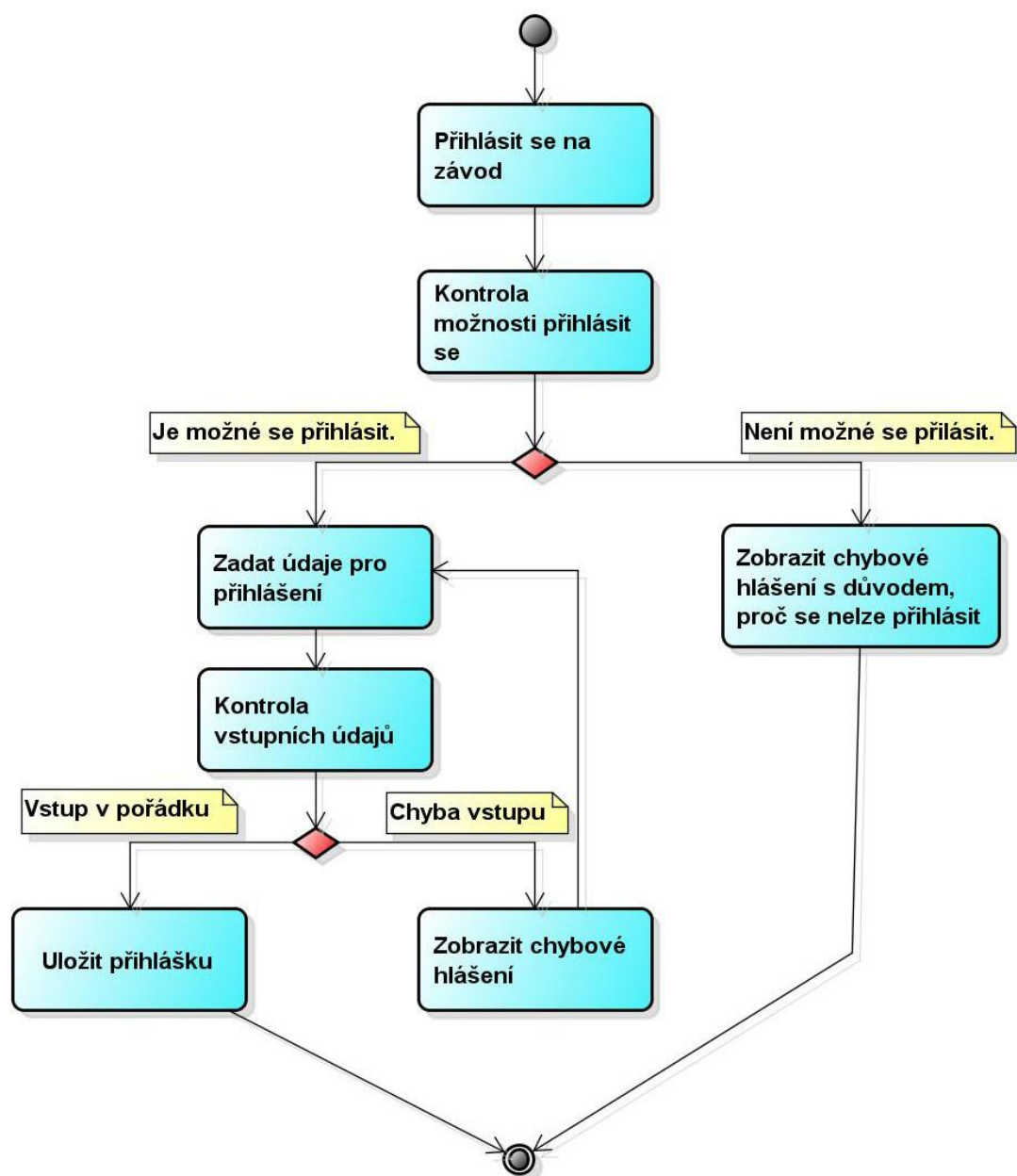
Následuje akce *Kontrola možnosti přihlásit se*, která směřuje do rozhodovacího uzlu se dvěma variantami vyhodnocení.

Pokud není možné se přihlásit, následuje akce *Zobrazit chybové hlášení s důvodem, proč se nelze přihlásit* a případ užití tím končí.

Pokud je možné se přihlásit, následují akce *Zadat údaje pro přihlášení* a *Kontrola vstupních údajů*, která směřuje do dalšího rozhodovacího uzlu se dvěma variantami vyhodnocení.

Pokud jsou vstupní údaje chybné, následuje akce *Zobrazit chybové hlášení*, po jejímž vykonání se případ užití vrací zpět k akci *Zadat údaje pro přihlášení*.

Pokud jsou vstupní údaje v pořádku, provede se akce *Uložit přihlášku* a případ užití tím končí.



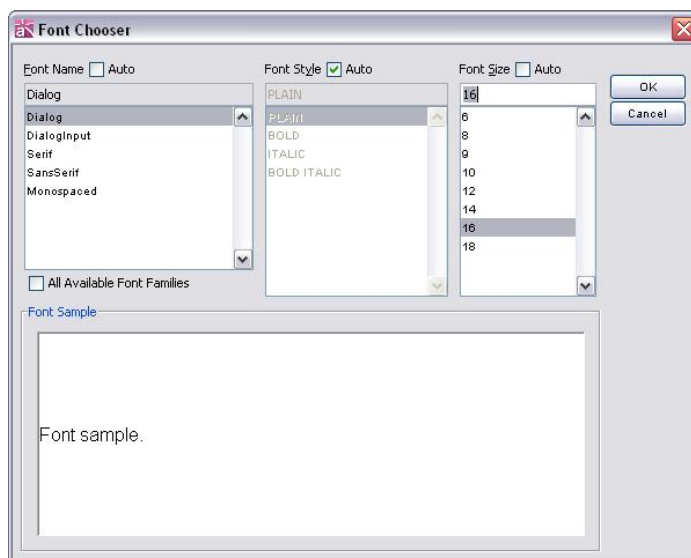
Obrázek 3.17 aktivit Přihlásit se na závod, zdroj: autor dle [3]

3.4.2 Postup tvorby diagramu aktivit v Astah* professional

Diagram aktivit hlavního byznys procesu aplikace OB ZAVODY, zachyceném na obrázku 3.16, můžete v programu Astah* professional vytvořit v následujících krocích:

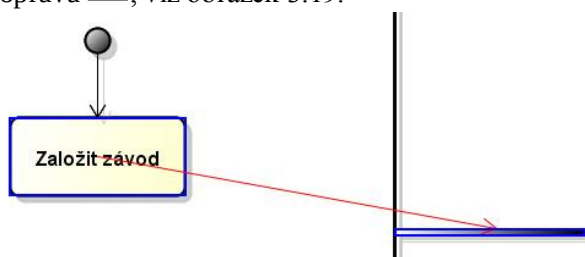
1. V projektu OBzavody, který jste vytvořili pro diagram případů užití (viz kapitola 3.2.2) založte nový balíček s názvem *Activity Diagram*. Postup založení balíčku v projektu je popsán v kapitole 2.2.3.
2. Do balíčku *Activity Diagram* vložte diagram aktivit (viz kapitola 2.2.3).
V editoru diagramu se zobrazí okno *Activity Diagram 0/ Activity Diagram [Activity Diagram]* pro návrh diagramu aktivit, dále jen diagram aktivit.
3. Pro lepší čitelnost diagramu můžete zvětšit písmo:
 - v hlavní nabídce zvolte [Tool] – [Project] – [Default Font...],

- ve vyvolaném dialogovém okně *Font Chooser* odškrtněte *Font Size [Auto]* a změňte velikost písma např. na 16, viz obrázek 3.18.



Obrázek 3.18 Nastavení fontu písma, zdroj: prostředí programu Astah* professional

- Do diagramu aktivit vložte dvě svislé dráhy – volba svislého obdélníku .
- Pokud chcete dráhy na první pohled odlišit, můžete je různě obarvit. V diagramu aktivit na obrázku 3.16 je například levá dráha červená a pravá dráha žlutá.
Z lokální nabídky dráhy vyberte [*Set Color...*], což vyvolá okno *Color Chooser* pro nastavení barvy, viz kapitola 2.2.9.
- Do levé dráhy umístěte počáteční uzel – volba plného kolečka .
- Pod počáteční uzel vložte aktivitu *Založit závod* – volba šestiúhelníku .
- Do levé části pravé dráhy umístěte rozvětvení – volba čáry, do které jedna šipka vchází a dvě z ní vychází .
- Do levé dráhy umístěte aktivitu *Vypsát pořádané závody* a do pravé dráhy na stejnou úroveň aktivitu *Vypsát závody*.
- Na stejnou úroveň do obou drah umístěte rozhodování – volba kosočtverce .
- Do levé dráhy pod rozhodování vložte aktivity *Upravit informace o závodě*, *Upravit seznam přihlášených* a *Zrušit závod*. Do pravé dráhy pod rozhodování na stejnou úroveň umístěte aktivity *Přihlásit se na závod*, *Upravit přihlášení na závod* a *Odhlásit se ze závodu*.
- Na stejnou úroveň do obou drah vložte koncový uzel – volba plného kolečka, ohraničeného bílou linkou .
- Prvky diagramu spolu propojte šipkou dle vzoru finálního obrázku 3.16 – volba šipky doprava , viz obrázek 3.19.



Obrázek 3.19 Propojení prvků diagramu šipkou, zdroj: prostředí programu Astah* professional

14. Nakonec prvky diagramu přehledně uspořádejte a doporučuji i sjednotit velikosti aktivit, stejným způsobem jako při úpravě případů užití, viz kapitola 3.2.2.
15. Hotový diagram aktivit můžete vyexportovat jako obrázek do formátu JPEG, viz kapitola 2.2.7.

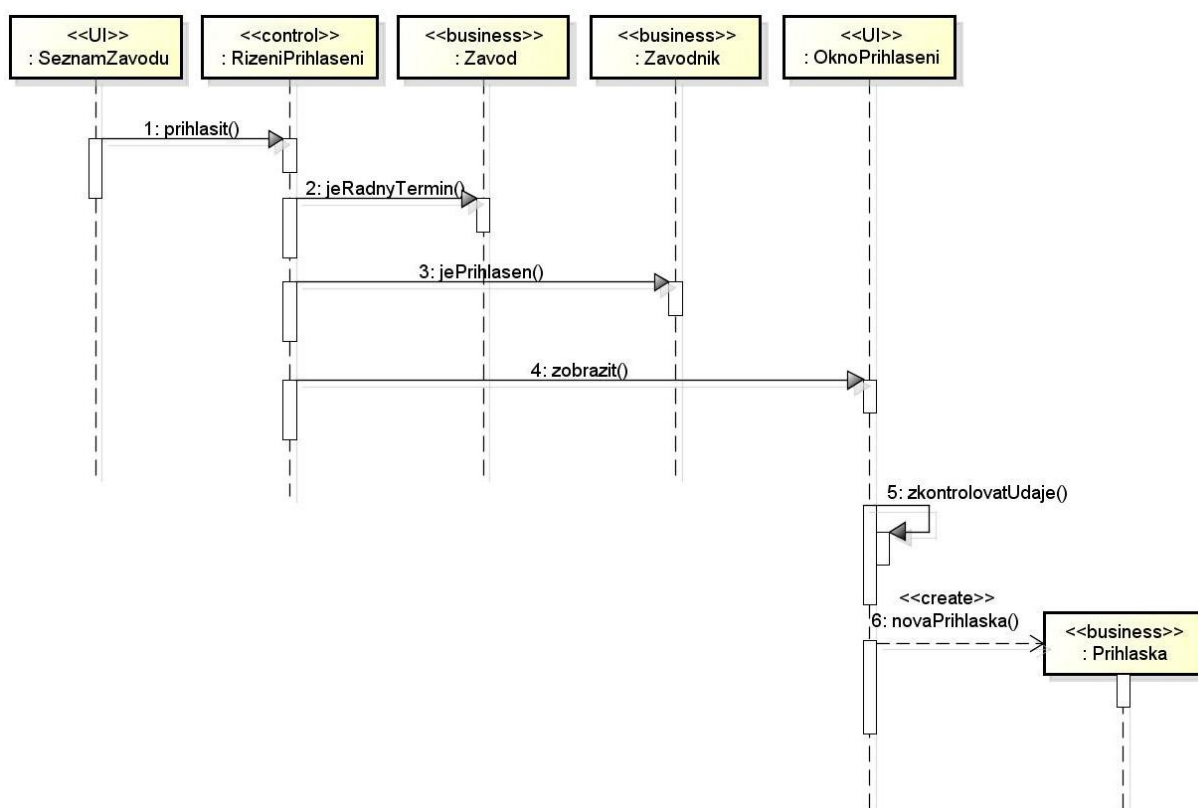
3.5 Sekvenční diagram

Sekvenční diagram (Sequence diagram) se řadí do skupiny diagramů interakcí. Zachycuje časově uspořádanou posloupnost zaslání zpráv mezi objekty. [3]

Sekvenční diagram nejčastěji znázorňuje spolupráci několika vzorových objektů v rámci jednoho případu užití. [4]

Prvky sekvenčního diagramu jsou klasifikátory (dle [4] také účastníci), jejich čáry života a zprávy, které si mezi sebou posílají.

Podrobnou charakteristiku sekvenčního diagramu a jeho prvků naleznete např. v [4].



Obrázek 3.20 Sekvenční diagram případu užití Přihlásit se na závod, zdroj: autor dle [3]

3.5.1 Sekvenční diagram aplikace OB ZAVODY

Na obrázku 3.20 je zachycen sekvenční diagram, který znázorňuje interakci objektů, vycházející ze specifikace případu užití *UC5 Přihlásit se na závod*.

Následuje popis jednotlivých zpráv mezi objekty v časové posloupnosti:

1: prihlasit()

Sekvence zpráv je zahájena, když uživatel u závodu vybraného ze seznamu závodů zadá volbu *přihlásit se na závod*. Objekt třídy typu <<uživatelské rozhraní>> *SeznamZavodu* pošle požadavek na přihlášení objektu třídy typu <<control>> *RizeniPrihlaseni*.

2: jeRadnyTermin() a 3: jePrihlasen()

Objekt třídy *RizeniPrihlaseni* kontroluje, zda je přihlášení možné. Nejprve zasílá zprávu *jeRadnyTermin()* objektu třídy typu <<business>> *Zavod*, čímž prověřuje, jestli již nevypršel termín řádných přihlášek. Potom posílá zprávu *jePrihlasen()* objektu třídy typu <<business>> *Zavodnik* a zjišťuje, zdali už závodník není přihlášen.

4: zobrazit()

V případě, že přihlášení je možné, objekt třídy *RizeniPrihlaseni* odešle objektu třídy typu <<uživatelské rozhraní>> *OknoPrihlaseni* požadavek na zobrazení okna, ve kterém uživatel vyplní údaje potřebné pro přihlášení na závod.

5: zkontrolovatUdaje()

Po vyplnění údajů pro přihlášení na závod a jejich potvrzení objekt třídy *OknoPrihlaseni* zkontroluje správnost zadaných údajů.

6: novaPrihlaska()

Pokud jsou zadané údaje v pořádku, objekt třídy *OknoPrihlaseni* vytvoří nový objekt třídy typu <<business>> *Prihlaska*.

3.5.2 Postup tvorby sekvenčního diagramu v Astah* professional

Sekvenční diagram případu užití *UC5 Přihlásit se závod*, zachycený na obrázku 3.20, můžete v programu Astah* professional vytvořit v následujících krocích:

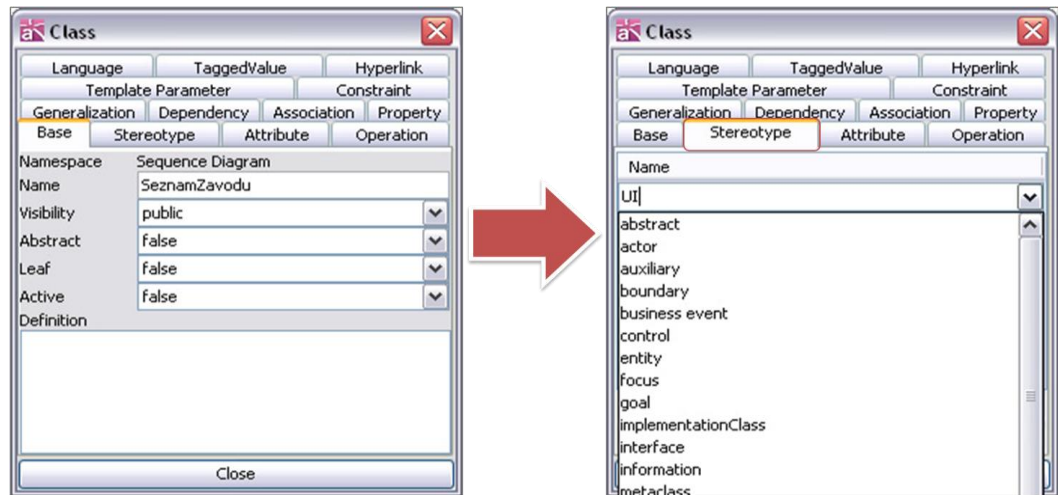
1. V projektu OBzavody, který jste vytvořili pro diagram případů užití (viz kapitola 3.2.2) založte nový balíček s názvem *Sequence Diagram*. Postup založení balíčku v projektu je popsán v kapitole 2.2.3.
2. Do balíčku *Sequence Diagram* vložte diagram aktivit (viz kapitola 2.2.3).
V editoru diagramu se zobrazí okno *Sequence Diagram 0/ Sekvence Diagram [Sequence Diagram]* pro návrh sekvenčního diagramu, dále jen sekvenční diagram.
3. Pro lepší čitelnost diagramu můžete zvětšit písmo. Postup je stejný jako v kapitole 3.4.2 u digramu aktivit.
4. Do sekvenčního diagramu vložte a umístěte vedle sebe 6 klasifikátorů (v tomto případě objektů) s jejich čarami života – volba obdélníku s přerušovanou čarou .
5. Existují různé varianty, jak můžete objekt v sekvenčním diagramu označit:
 - a) pouze názvem objektu, např. *ObjektA*,
 - b) názvem objektu včetně názvu třídy, např. *ObjektA: TridaA*,
 - c) název objektu uveden není a je určena pouze třída, např. *:TridaA*.

Objekt se v sekvenčním diagramu pojmenovává, pokud jej např. chcete předat jako parametr zprávy. [3] Většinou ale stačí objekt označit pouze určením třídy (varianta c). Objekty v sekvenčním diagramu proto doporučuji nazvat⁶ dle varianty c): *:SeznamZavodu*, *:RizeniPrihlaseni*, *:Zavod*, *:Zavodnik*, *:OknoPrihlaseni* a *:Prihlaska*

6. U sekvenčních diagramů se pro doplnění informace doporučuje klasifikátory obohatit o stereotypy. [3] Následuje postup přidání stereotypu do klasifikátoru (objektu):
 - označte klasifikátor (objekt),
 - v *Property View* na záložce *Base* klikněte na tlačítko *[Property]*,
 - zobrazí se dialogové okno s názvem *Class* jako na obrázku 3.21 nalevo,

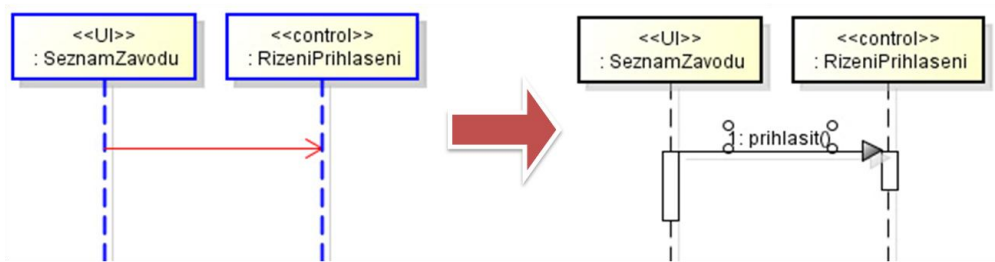
⁶ Při zadání názvu objektu tímto způsobem se program zeptá, zda chcete název za dvojtečkou vytvořit jako nový typ, což je v pořádku. Hlášení programu potvrďte.

- v okně *Class* přepněte na záložku *Stereotype* a klikněte na tlačítko *[Add]*,
- objeví se rozbalovací menu, ve kterém si můžete vybrat některý ze stereotypů, ale máte zde možnost zadat i stereotyp vlastní, viz obrázek 3.21 napravo.



Obrázek 3.21 Přidání stereotypu, zdroj: prostředí programu Astah* professional

7. Objekty mezi sebou propojte synchronními zprávami⁷ dle vzoru finálního obrázku 3.20 – volba šipky s písmenem S , viz obrázek 3.22.



Obrázek 3.22 Synchronní zpráva, zdroj: prostředí programu Astah* professional

8. Objekt *OknoPrihlaseni* propojte s objektem *Prihlaska* zprávou opatřenou stereotypem *<<create>>* – volba přerušované šipky s písmenem C , která reprezentuje vytvoření objektu *Prihlaska*.
9. Hotový sekvenční diagram můžete vyexportovat jako obrázek do formátu JPEG, viz kapitola 2.2.7.

⁷ Synchronní zpráva na rozdíl od zprávy asynchronní čeká na ukončení zpracování zprávy.[3]

4. ZÁVĚR

Hlavním cílem této práce bylo ukázat použití diagramů UML 2.0 na konkrétním příkladě a popsat postup jejich tvorby v CASE nástroji Astah*. Pro splnění těchto cílů byly nejprve nadefinovány požadavky na konkrétní aplikaci, spolu se sestavením slovníčku pojmů dané problémové oblasti a v souladu s nimi byl následně navržen ukázkový analytický model aplikace. Tento model tvoří čtyři typy UML diagramů, které výstižně zachycují problémovou oblast řešení: diagram případů užití, diagram tříd, diagram aktivit a sekvenční diagram. Kromě samotného návrhu diagramů, je u každého z nich dále uvedena stručná obecná charakteristika, vysvětlena jeho logika a popsány konkrétní kroky postupu jeho vytvoření v nástroji Astah* professional.

Diagram případů užití (Use Case Diagram) zachycuje chování systému z hlediska uživatele. Prakticky je na něm ukázán a objasněn kromě interakcí aktérů s případy užití i vztah generalizace/specializace mezi aktéry a vztah include mezi případy užití. Model případů užití se skládá nejen z diagramu případů užití ale i ze specifikací jednotlivých případů užití. Na ukázkou byla pro jeden případ užití specifikace zkompletována včetně návodu na její vytvoření v Astah* professional.

Diagram tříd (Class Diagram) zachycuje strukturu objektových tříd a jejich vzájemné vztahy v systému.[6] Prakticky je na něm demonstrován vztah generalizace/specializace mezi třídami, vztah kompozice, vztah běžné asociace a vztah asociace s asociací třídou.

Diagram aktivit (Activity Diagram) zobrazuje posloupnost aktivit, které mohou probíhat jak sekvenčně, tak paralelně.[3] Praktické použití tohoto diagramu je ukázáno jednak při zachycení hlavního byznys procesu definované aplikace a poté při znázornění průběhu vybraného případu užití.

Sekvenční diagram (Sequence diagram) zachycuje časově uspořádanou posloupnost zasílání zpráv mezi objekty. [3] V práci je tento diagram použit pro znázornění interakce objektů v rámci vybraného případu užití.

Postup vytváření těchto diagramů v Astah* professional podrobně popisuje jednotlivé kroky, jak daný diagram v nástroji vytvořit. Kromě klíčových kroků postupu, jsou zde zahrnuta i cenná doporučení, která mohou přispět především k vizuálnímu vylepšení diagramu.

Dalším cílem práce bylo stručně charakterizovat CASE nástroj Astah* a popsat základy práce v tomto programu. Ještě před samotnou charakteristikou nástroje Astah* je nejprve objasněn pojem CASE nástroj. Stěžejní část charakteristiky nástroje Astah* potom zahrnuje především popis možností všech jeho vydávaných edicí: Astah* professional, Astah* UML, Astah* Community a Astah* Share. Pro zpracování veškerých návodů a postupů v této práci byla zvolena edice Astah* professional, jelikož poskytuje komplexní funkcionalitu. Popis práce v Astah* professional se zaměřuje převážně na základní funkce programu, které jsou pro uživatele nejvíce potřebné. Popsány jsou například možnosti zobrazení, možnosti tisku, export obrázku a další.

Jazyk UML poskytuje mnoho různých technik a záleží jen na analytikovi, které z nich si pro analýzu a návrh dané aplikace vybere. Doporučení, na kterém se ale všichni autoři publikací o UML modelování shodují, je vytvářet modely jednoduché a srozumitelné. Za hlavní přínos této práce proto pokládám vytvoření jednoduchých a snadno pochopitelných UML modelů, které dokumentují konkrétní aplikaci. Dalším přínosem práce je zpracování návodů pro práci v programu Astah* professional, ve kterých je na rozdíl od obsáhlých manuálů popsána jen ta funkčnost programu, kterou uživatel nejvíce upotřebí.

Práce je koncipována jako příručka, která bude sloužit především jako studijní materiál pro výuku kurzu softwarového inženýrství. Je zaměřena výhradně na tvorbu UML diagramů na konceptuální úrovni, což je vzhledem k vysvětlení logiky diagramů a popisu kroků postupu jejich tvorby zcela postačující. Jako námět pro rozšíření modelu nadefinované aplikace však doporučuji podrobnější analýzu a návrh této aplikace na designové úrovni. Dalším námětem pro rozšíření je případné zařazení dalších typů UML diagramů do modelu.

Výsledky práce budou publikovány na internetu, kde budou k dispozici nejen studentům, ale i všem ostatním zájemcům o problematiku modelování v UML.

5. SEZNAM POUŽITÉ LITERATURY

5.1 Knižní zdroje

- [1] ARLOW, Jim; NEUDSTAT, Ila. *UML 2 a unifikovaný proces vývoje aplikací : Objektově orientovaná analýza a návrh prakticky*. 1. vyd. Brno : Computer Press, a.s., 2007. 567 s. ISBN 978-80-251-1503-9.
- [2] BUCHALCEVOVÁ, Alena; STANOVSKÁ, Iva; ŠIMŮNEK, Milan. *Základy softwarového inženýrství : základní témata*. 1. vyd. Praha: Vysoká škola ekonomická : Oeconomica, 2002. 198 s. ISBN 80-245-0346-8.
- [3] BUCHALCEVOVÁ, Alena; PAVLÍČKOVÁ, Jarmila; PAVLÍČEK, Luboš. *Základy softwarového inženýrství : materiály ke cvičení*. 1. vyd. Praha: Vysoká škola ekonomická : Oeconomica, 2007. 222 s. ISBN 978-80-245-1270-9.
- [4] FOWLER, Martin. *Destilované UML*. První vydání. Praha 7 : Grada Publishing, a.s., 2009. 176 s. ISBN 978-80-247-2062-3.
- [5] GÁLA, Libor ; POUR, Jan; ŠEDIVÁ, Zuzana. *Podniková informatika*. Druhé vydání. Praha 7 : Grada Publishing, a.s., 2009. 496 s. ISBN 978-80-247-2615-1.
- [6] KANISOVÁ, Hana; MÜLLER, Miroslav. *UML srozumitelně*. 2. aktualizované vydání. Brno : Computer Press, a.s., 2007. 176 s. ISBN 80-251-1083-4.

5.2 Internetové zdroje

- [7] ABZ slovník cizích slov [online]. c2005-2006 [cit. 2011-04-27]. Dostupné z WWW: <<http://slovník-cizich-slov.abz.cz/>>.
- [8] *Astah** [online]. c2009 [cit. 2011-02-22]. *Astah* Basic Operation Guide*. Dostupné z WWW: <http://Astah.change-vision.com/en/files/Astah_Basic_Operation_Guide-e.pdf>.
- [9] *Astah** [online]. c2011 [cit. 2011-02-14]. *Astah* community - Free UML Modeling Tool*. Dostupné z WWW: <<http://Astah.change-vision.com/en/>>.
- [10] *Astah** [online]. c2011 [cit. 2011-02-14]. *Astah* community - Free UML Modeling Tool*. Dostupné z WWW: <<http://Astah.change-vision.com/en/product/Astah-community.html>>.
- [11] *Astah** [online]. c2011 [cit. 2011-02-14]. *Astah* professional - Software Design Tool*. Dostupné z WWW: <<http://Astah.change-vision.com/en/product/Astah-professional.html>>.
- [12] *Astah** [online]. c2011 [cit. 2011-02-14]. *Astah* share - View Astah* diagram on the Web*. Dostupné z WWW: <<http://Astah.change-vision.com/en/product/Astah-share.html>>.
- [13] *Astah** [online]. c2011 [cit. 2011-02-14]. *Astah* UML, modeler with Mind Map*. Dostupné z WWW: <<http://Astah.change-vision.com/en/product/Astah-uml.html>>.
- [14] *Astah** [online]. c2011 [cit. 2011-02-14]. *Astah* UML pad - Modeling UML Anytime, Anywhere on iPad*. Dostupné z WWW: <<http://Astah.change-vision.com/en/product/Astah-pad.html>>.
- [15] *Astah** [online]. c2011 [cit. 2011-02-14]. *License and Price*. Dostupné z WWW: <<http://Astah.change-vision.com/en/license.html#pro-user>>.
- [16] *Astah** [online]. c2011 [cit. 2011-02-14]. *Why did you change the product name from JUDE to Astah*?*. Dostupné z WWW: <<http://Astah.change-vision.com/en/faq/Astah-faq/change-product-name.html>>.
- [17] *Astah* community – Wikipedie, otevřená encyklopedie* [online]. St. Petersburg (Florida) : Wikimedia Foundation, 2001- , strana naposledy edit. 2010-11-04 [cit 2011-2-14]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Astah*>
- [18] BENEŠ, Michal ; PAVLÍČEK, Luboš. *Objektová analýza, návrh a programování* [online]. 2008 [cit. 2011-04-27]. *Objekty - Metodiky a Notace-termíny*. Dostupné z WWW: <<http://objekty.vse.cz/Objekty/MetodikyANotace-termíny>>.

- [19] *CASE nástroje – Wikipedie, otevřená encyklopedie* [online]. St. Petersburg (Florida) : Wikimedia Foundation, 2001- , strana naposledy edit. 2011-02-11 [cit 2011-2-18]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/CASE_n%C3%A1stroje>
- [20] *ČSOS - Hanácká oblast* [online]. c2011 [cit. 2011-03-06]. Přihlášky na závody. Dostupné z WWW: <<http://www.obhana.cz/prihlasky.asp>>
- [21] *Oficiální stránka sekce orientačního běhu Českého svazu orientačních sportů* [online]. c2002-2011 [cit. 2011-03-06]. Pravidla OB. Dostupné z WWW: <http://orientacnibeh.cz/ob/dokumenty/pravidla_ob.htm>
- [22] *Interakce – Wikipedie, otevřená encyklopedie* [online]. St. Petersburg (Florida) : Wikimedia Foundation, 2001- , strana naposledy edit. 2011-04-26 [cit 2011-04-27]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/Interakce>>
- [23] PAVLÍČEK, Luboš. *Kitscm.vse.cz* [online]. c2011, 2010-12-15 09:54 [cit. 2011-02-14]. Licence: Astah. Dostupné z WWW: <<https://kitscm.vse.cz/dokuwiki/doku.php?id=licence:Astah>>.
- [24] Referenční manuál z nápovědy
- [25] REJNKOVÁ, Petra. *Příklady použití diagramů UML* [online]. Praha : Katedra informačních technologií, 2009. 59 s. Bakalářská práce. Vysoká škola ekonomická v Praze. Dostupné z WWW: <http://www.vse.cz/vskp/public.php?evskp_id=13824>.

6. TERMINOLOGICKÝ SLOVNÍK

Termín	Zkratka	Význam [zdroj]
abstrakce		vědomé a účelové odhlédnutí od jistých aspektů problému [3]
aktér		část okolí vytvářeného systému, pro kterou je typická komunikace se systémem [3]
algoritmus		postup zpracování, jehož charakteristickými vlastnostmi jsou rezultativnost, diskretnost a hromadnost [3]
analýza		1) postup, jehož výstupem je logický model vytvářeného systému [3] 2) etapa vývoje systému (aplikace) [1]
asociace		vztah mezi třídami [3]
asociační třída		asociace, která udržuje vlastní atributy a metody [autor]
atribut		datová část objektu, tj. proměnná, která je součástí objektu a je definována ve třídě [3]
balíček		seskupení prvků modelu, které spolu souvisí [autor]
Computer Aided Software Engineering	CASE	nástroje podporující vývoj softwaru [3]
data		jakékoli fyzicky (materiálně) zaznamenané znalosti, poznatky, zkušenosti nebo výsledky pozorování procesů, projevů, činností a prvků reálného světa [2]
datový typ		typ dat uložených v proměnné [3]
design		návrh, etapa vývoje systému [3]
designová úroveň		vyjadřuje pohled na obsah systému, tzn. jaké funkce bude systém provádět, jaká data bude uchovávat, v konkrétním prostředí, ve kterém bude systém realizován [3]
diagram		určitý pohled na navrhovaný systém [3]
dialogové okno		druh okna, které nelze minimalizovat ani měnit jeho velikost [3]
element		prvek [autor]
entita		označení pro obecný a jasně definovaný prvek (např. třída, objekt, proces) [2]
forward engineering		generování struktury programového kódu nebo databáze na základě vytvořených diagramů (např. v CASE nástroji - automatické generování kódu) [18]
funkcionalita (softwaru)		soubor všech funkcí, které daný software poskytuje [autor]
generalizace/specializace		vytčení společných vlastností několika prvků do jednoho hierarchicky nadřazeného [3]
implementace		1) etapa návrhu systému [2] 2) realizace (naprogramování) systému [2]
implementační úroveň		vyjadřuje pohled na obsah systému, který je grafickým zachycením implementovaného kódu [3]

informační systém	IS	konzistentní uspořádaná množina komponent spolupracující za účelem tvorby, shromažďování, zpracování, přenášení a rozšiřování informací, přičemž prvky informačního systému tvoří lidé (uživatelé informací) a informatické zdroje [5]
instance		konkrétní výskyt abstraktní představy [1]
interakce		vzájemné působení [22]
klasifikátor		abstraktní vyjádření typu předmětu [1]
kompozice		druh asociace, který zachycuje vztah celku a části, přičemž část nemůže existovat samostatně bez celku [3]
komunikace		předávání údajů mezi objekty prostřednictvím zpráv nebo volání metod [3]
konceptuální úroveň		vyjadřuje pohled na obsah systému, tzn. jaké funkce bude systém provádět, jaká data bude uchovávat, nezávisle na prostředí, ve kterém bude systém realizován [2]
licence		oprávnění k používání (softwaru) [7]
lokální menu		místní nabídka, vztahuje se k určitému elementu [autor]
menu		nabídka, prvek uživatelského rozhraní [3]
metoda		funkční složka objektu, která zajišťuje jeho chování a má formu funkce nebo procedury [3]
model		1) dílčí pohled na vytvářený systém [3] 2) souhrn všech pohledů na celý systém [3]
modelování		proces tvorby modelů, které popisují navrhovaný systém [autor]
modul		část systému na dané úrovni podrobnosti již dále nerozkládaná, s relativně samostatnou funkčností [2]
nástrojová lišta		panel nástrojů, soubor tlačítek s ikonami pro rychlé vyvolání funkce [3]
návrh		etapa vývoje systému, ve které je upřesňován analytický model systému [3]
notace		forma zápisu, pro níž platí určité konvence [1]
objekt		programový nebo konstrukční prvek, který obsahuje data i procedury [2]
objektově orientovaný	OO	přístup, jehož základní myšlenkou je seskupit data a funkce, které s nimi pracují [2]
operace		obecná odpovědnost třídy ve fázi analýzy, která je nutná k pochopení modelu [1]
proces		posloupnost stavů a činností [autor]
procesní modelování		modelování firemních procesů [6]
programování		proces, jehož výsledkem je program pro počítačové zpracování [2]
repository		knihovna navržených prvků, obsahuje všechny vytvořené prvky ve všech modelech [18]
reverse engineering		zpětná transformace zdrojového kódu programu nebo

		struktury databáze do diagramů [autor]
role		prvek v diagramu tříd, který popisuje roli třídy v daném vztahu [3]
rozbalovací menu		druh menu, které se po kliknutí rozbalí [3]
software	SW	počítačový program [7]
stereotyp		mechanismus rozšíření standardních prvků definovaných v UML [2]
struktura		způsob uspořádání [autor]
šablona		předloha, která je společná pro určitý typ prvků, čímž usnadňuje jejich vytváření [autor]
třída		abstrakce objektů se stejnými vlastnostmi, chováním a vztahy k ostatním objektům [3]
Unified Modelling Language	UML	univerzální jazyk pro vizuální modelování systémů [1]
uživatel		člověk, který bude používat vytvářený systém [2]
uživatelské rozhraní	UI	část aplikace, která slouží pro interakci s uživatelem [2]
verze (programu)		plánovaná změna některých charakteristik aplikace [2]
view		pohled [autor]
webová aplikace		aplikace s tenkým klientem, běží na serveru a uživatel k ní přistupuje prostřednictvím webového prohlížeče [autor]
zadáání		předpis, jak má daný systém vypadat [2]
zpráva		způsob komunikace mezi objekty [2]