

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Diagramy aktivit

Aplikace pro tvorbu diagramů aktivit v notaci UML



2018

Vedoucí práce: RNDr. Arnošt Ve-
čerka

Robin Prinke

Studijní obor: Aplikovaná informatika,
kombinovaná forma

Bibliografické údaje

Autor: Robin Prinke
Název práce: Diagramy aktivit (Aplikace pro tvorbu diagramů aktivit v notaci UML)
Typ práce: bakalářská práce
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby: 2018
Studijní obor: Aplikovaná informatika, kombinovaná forma
Vedoucí práce: RNDr. Arnošt Večerka
Počet stran: 32
Přílohy: 1 CD/DVD
Jazyk práce: český

Bibliographic info

Author: Robin Prinke
Title: Activity diagrams (Application to drawing Activity Diagrams in UML Notation)
Thesis type: bachelor thesis
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense: 2018
Study field: Applied Computer Science, combined form
Supervisor: RNDr. Arnošt Večerka
Page count: 32
Supplements: 1 CD/DVD
Thesis language: Czech

Anotace

Práce je stručnou dokumentací tvorby aplikace pro tvorbu diagramu aktivit v notaci UML. Obsahuje popis nejčastějších prvků diagramu, dále základní technické informace k aplikaci a v poslední řadě seznámení s ovládáním aplikace pro uživatele.

Synopsis

The work is a brief documentation of the creation process of an application for creating an activity diagram in the UML notation. It contains a description of the most common elements of the diagram, basic technical information about the application and, lastly, familiarity with the application control for the user.

Klíčová slova: UML; diagram aktivit; aplikace; tvorba diagramu; WPF; C#

Keywords: UML; activity diagrams; application; drawing diagram; WPF; C#

Na tomto místě bych chtěl poděkovat všem, kteří mi byli při zpracování bakalářské práce nápomocni. Obzvláště RNDr. Arnoštu Večerkovi za cenné připomínky, kterými přispěl k vypracování této práce. Nemalé poděkování patří také mé přítelkyni za velkou trpělivost, kterou se mnou projevila v průběhu celého mého studia a tvorbě této práce.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	UML	8
1.1	Diagram aktivit	9
1.2	Uzly diagramu	9
1.2.1	Vykonatelné uzly	9
1.2.1.1	Akce vykonávající	9
1.2.1.1.1	Akce	10
1.2.1.1.2	Odeslání signálu	10
1.2.1.2	Akce přijímající události	10
1.2.1.2.1	Přijetí signálu	10
1.2.1.2.2	Časová událost	11
1.2.2	Objektové uzly	11
1.2.2.1	Objekt	11
1.2.2.2	Pin	11
1.2.3	Řídící uzly	12
1.2.3.1	Počátek	12
1.2.3.2	Konec toku	12
1.2.3.3	Konec aktivity	13
1.2.3.4	Rozhodnutí	13
1.2.3.5	Sloučení	13
1.2.3.6	Rozvětvení	13
1.2.3.7	Sjednocení	14
1.3	Vztahy mezi prvky	15
1.3.1	Hrany	15
1.3.1.1	Řídící tok	15
1.3.1.2	Objektový tok	15
1.3.1.3	Přerušeni	15
1.3.2	Oddíly	16
1.3.3	Oblasti	16
2	Zadání a požadavky	17
2.1	Příklady programů	17
2.2	Požadované funkce a vlastnosti aplikace	18
3	Technická dokumentace	19
3.1	Technologie	19
3.2	Struktura programu	19
3.2.1	ActivityDiagram.AppDiagram	19
3.2.2	ActivityDiagram.AppWindows	20
3.2.3	ActivityDiagram.GraficClass	20
3.2.4	ActivityDiagram.HelperFactory	21
3.2.5	ActivityDiagram.Notification	21
3.3	Popis třídy Grafics	21
3.3.1	GaficsObject	21

3.3.1.1	GONode	22
3.3.1.2	GOArea	22
3.3.2	GraficConnector	23
4	Uživatelská dokumentace	23
4.1	Instalace aplikace	23
4.2	Ovládací prvky	24
4.2.1	Hlavní okno aplikace	24
4.2.2	Panel pro vkládání prvků diagramu	24
4.3	Práce s grafickými prvky diagramu	25
Závěr		27
4.4	Zhodnocení a náměty	27
Conclusions		28
4.5	Assessments and Suggestions	28
A	Příloha č.1	29
B	Příloha č.2	30
C	Obsah přiloženého CD/DVD	31
Literatura		32

Seznam obrázků

1	Akce	10
2	Odeslání signálu	10
3	Přijetí signálu	11
4	Časová událost	11
5	Uzel objektu	12
6	Uzel akce s objektovými piny	12
7	Uzel počátku	12
8	Uzel konce toku	12
9	Uzel konce aktivity	13
10	Uzel rozhodnutí	13
11	Uzel sloučení	13
12	Příklad rozhodnutí	14
13	Uzel rozvětvení	14
14	Uzel sjednocení	14
15	Příklad paralelního toku	14
16	Řídící toky	15
17	Objektové toky	16
18	Přerušení	16
19	Plavecká dráha	17
20	Oblast - region	17
21	Hlavní okno aplikace	25
22	Panel pro vkládání prvků diagramu	25
23	Zobrazení struktury programu	29
24	Zobrazení dědičnosti třídy Graphics	30

1 UML

UML (Unified Modeling Language) je standardní vizuální modelovací jazyk, který je používán [7]:

- pro modelování procesů,
- analýzu, návrh a vývoj (informační) systémů.

UML slouží analytikům, architektům i vývojářům k popisu, návrhu a dokumentaci různých procesů a také struktur a chování (informačních) systémů. Zároveň je to jazyk univerzální [2], protože není závislý na:

- **doméně** (používá se jak například v energetice, tak například v bankovníctví),
- **vývojové metodologii** (používá se jak při vodopádovém, tak agilním způsobu vývoje),
- **technologii** (používá se jak při vývoji v PHP a MySQL, tak při vývoji v C#a MS SQL

UML je důležité chápat pouze jako nástroj [2], který využívá obecný proces vývoje software, který, a to i za pomoci UML, teprve:

- poskytuje poradenství ohledně pořadí a náplně činností týmu,
- určuje, jaké artefakty by měly být dodány,
- řídí úkoly jednotlivců i týmu jako celku,
- navrhuje kritéria pro monitorování a měření výstupů vývojového projektu.

Diagram UML je částečné grafické zobrazení modelu systému (ať již pouze navrženého nebo existujícího). Tento diagram zachycuje zpravidla dva základní aspekty modelu [8], které představují prvky v UML modelu:

- prvky (tzv. Elements),
- vztahy (tzv. Relations).

UML notace [8] je přitom poměrně volná, protože model systému může obsahovat i jinou dokumentaci, jako jsou případy použití napsané jako šablony nebo nevylučuje kombinaci různých druhů diagramů v jednom tak, aby zobrazovaly model co nejdůležitěji a pro uživatele nejčitelněji (proto nejsou hranice mezi diagramy striktně vynucovány s výjimkou některých nástrojů pro tvorbu UML, které toto přímo neumožňují).

Zároveň je důležité mít na paměti, že UML není zcela komplexní pro předávání informací [5]. Vizuální základ v UML modelu je potřeba dostatečně přesně vysvětlit například popisem slovním, právě i z důvodu určité volnosti notace

UML (diagram může být dvěma osobami vyložen dvě různými způsoby, protože kromě přesných notačních principů poskytuje i určitou tvůrčí volnost).

Podle UML notace [8] jsou definovány dvě základní sady diagramů

- **strukturální**, které popisují / modelují statickou strukturu systémů včetně jeho částí (především vztahy mezi jednotlivými částmi a prvky), a to na různé úrovni abstrakce a implementace,
- **behaviorální**, které popisují dynamické prvky systémů, tedy jako chování systému jako celku v čase, případně jeho částí (včetně interakcemi mezi nimi).

1.1 Diagram aktivit

Diagram aktivit je UML diagram, který patří do skupiny behaviorálních diagramů a slouží ke znázornění algoritmů a procesů. Modeluje toky v systémech, a to především s důrazem na následnost a podmíněnost jednotlivých kroků znázorňovaných v daném modelu. Tyto toky mohou být buď řídicího charakteru (control flow) nebo materiálního (object flow) [8].

Zjednodušeně je možné uvést, že diagram zobrazuje Aktivitu. Aktivita je podle UML parametrizované chování, které je reprezentováno jako graf uzlů spojených hranami.

Alternativou pro UML notaci zobrazování aktivit (a procesů) je BPMN (Business Process Model and Notation). Ve srovnání s UML poskytuje širší možnosti pro znázornění závislostí v diagramu, protože obsahuje více rozlišení pro samotné grafické prvky (UML obsahuje merge node, ale BPMN obsahuje 7 různých bran (gateways) pro upřesnění o jaký druh merge se jedná).

1.2 Uzly diagramu

V rámci diagramu aktivit jsou rozlišovány tři skupiny uzlů, které jsou popsány níže [8]. V rámci práce je výčet zjednodušen na nejběžnější uzly a uzly, které mají shodné znázornění nebo nejsou běžně potřebné, jsou vynechány. Zájemce se však s celkovým přehledem může seznámit přímo v notaci UML. Následný výčet je inspirován postupem tvorby diagramu [5]

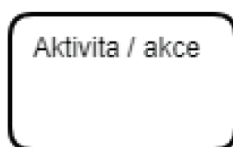
1.2.1 Vykonatelné uzly

Vykonatelné uzly (Executable nodes) vyjadřují kroky aktivity nižší úrovně. Výčet těchto uzlů je podle notace UML poměrně široký (Invocation Actions, Link Actions, Link Object Actions, Structural Feature Actions, Accept Event, Variable Actions, Structured Actions, Object Actions). Pro pochopení a praktické použití jsou ale nejdůležitější následující vybrané:

1.2.1.1 Akce vykonávající (Invocation Actions) jsou akce, které samo o sobě vytvářejí - vyvolávají nějaké změny v rámci chování modelovaného systému.

1.2.1.1.1 Akce (Action) je základní jednotka spustitelné funkce. Je obsažena přímo nebo nepřímo v modelu chování systému, které je modelován. Provedení akce představuje nějakou transformaci nebo zpracování v modelovaném systému, jímž může být informační systém. Provádění akce může vést k vyvolání jiného chování například další akce.

Grafická notace viz obrázek 1 - obdélník se zaoblenými rohy obsahující označení / název Akce ve specifickém tvaru spojení podstatného jména (předmětu) a slovesa (aktivity s předmětem) například „Vytvořit objednávku“.



Obrázek 1: Akce

1.2.1.1.2 Odeslání signálu (Send signal action) je specifická akce, které vytváří a odesílá signál nebo zprávu. Tímto signálem může být myšlena například informace o dokončení nějaké akce, výjimka vyvolaná v programu apod.

Grafická notace viz obrázek 2 - pětihranný objekt (obdélník, který je na pravé straně rozšířen do tvaru šipky) obsahující označení odeslaného signálu například zprávy, výjimky apod.



Obrázek 2: Odeslání signálu

1.2.1.2 Akce přijímající události (Accept Event Actions) jsou akce, které zpracovávají změny nebo události řízené a vznikající v rámci i mimo modelovaný systém.

1.2.1.2.1 Přijetí signálu (Accept Event Action) je specifická akce, které zpracovává signál nebo zprávu. Tímto signálem může být myšlena například informace o dokončení nějaké akce, výjimka vyvolaná v programu apod. Tok aktivity v tomto místě začíná nebo je přeručeno do zachycením tohoto signálu nebo zprávy.

Tento uzel je úzce spojen s předchozím uzlem „Přijetí signálu“. Bohužel v originální anglické notaci je jednou uvedeno Signal, podruhé Event, což může znesnadnit pochopení souvislosti.

Grafická notace viz obrázek 3 - pětihranný objekt (obdélník, který je na levé straně vyříznut do tvaru šipky) obsahující označení přijímaného signálu například zprávy, výjimky apod.



Obrázek 3: Přijetí signálu

1.2.1.2.2 Časová událost (Accept Time Event Action) je specifická akce, která zpracovává časovou událost. Typicky je touto událostí uplynutí určeného času. Tok aktivity dále nepokračuje před jeho zpracováním, čímž je myšleno uplynutí časového úseku uvedeného u uzlu.

Grafická notace viz obrázek 4 - je znázorňována obrazcem čtvercového půdorysu, který je z obou stran vyříznut a připomíná tedy přesýpací hodiny.



Obrázek 4: Časová událost

1.2.2 Objektové uzly

Objektové uzly (object nodes) znázorňují prvky v diagramu, které obsahují určitá data vyjadřující vstup nebo výstup ze spustitelných uzlů. V rámci notace UML je rozlišováno dle charakteru dat více podtypů těchto uzlů. Pro zjednodušení použitelné pro tuto práci lze uvádět tyto dvě skupiny:

1.2.2.1 Objekt, což je pro tuto práci znázornění podle notace UML následujících uzlů: `ActivityParameterNodes`, `CentralBufferNodes` and `DataStoreNodes`. Mají velmi blízkou grafickou notaci, která se liší konkrétním umístěním v diagramu a použitím stereotypů v textových upřesněních u daného uzlu.

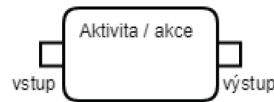
Grafická notace viz obrázek 5 - obdélník s ostrými rohy obsahující označení / název objektu například „Vytvořit objednávku“ - případně další upřesňující atributy (stereotyp, stavy objektu apod.).

1.2.2.2 Pin zobrazující objekt vstupu nebo výstupu z konkrétní akce, tento prvek je tedy pevně spojen s prvkem Akce a objektovým tokem v diagramu.

Grafická notace viz obrázek 6 - znázorňuje se malými čtverci, které jsou připojeny k prvku Akce a jsou textově označeny názvem objektu.



Obrázek 5: Uzel objektu



Obrázek 6: Uzel akce s objektovými piny

1.2.3 Řídící uzly

Řídící uzly (Control nodes) specifikují následnost vykonatelných uzlů v rámci řídicího toku aktivity, ve specifických případech výjimečně i vztahy v rámci objektového toku.

1.2.3.1 Počátek (Initial node) je uzel, který označuje počátek aktivity / diagramu, v jehož rámci se nachází vždy pouze jeden a vede z něj hrana.

Grafická notace viz obrázek 7 - kružnice a soustředný černý kruh menší velikosti, může být připojeno textové označení specifikace počátku.



Obrázek 7: Uzel počátku

1.2.3.2 Konec toku (Flow final node) je uzel, který označuje konec vybraného toku v diagramu, neoznačuje však absolutní konec celé aktivity.

Grafická notace viz obrázek 8 - kružnice a uvnitř obrazec ve tvaru písměně „X“, může být připojeno textové označení specifikace uzlu.



Obrázek 8: Uzel konce toku

1.2.3.3 Konec aktivity (Activity final node) je uzel, který označuje absolutní konec celé aktivity, jeho dosažením jsou ukončeny i paralelní toky diagramu.

Grafická notace viz obrázek 9 - černý kruh, může být připojeno textové označení specifikace uzlu.



Obrázek 9: Uzel konce aktivity

1.2.3.4 Rozhodnutí (Decision node) je uzel, který označuje v diagramu místo, kde se tok rozděluje a pokračuje pouze ve větvi, které splňuje definovanou podmínku tímto uzlem.

Grafická notace viz obrázek 10 - čtverec otočený o úhel 45 stupňů tzn. stojící na vrcholu, může být připojeno textové označení specifikace uzlu.



Obrázek 10: Uzel rozhodnutí

1.2.3.5 Sloučení (Merge node) je uzel, který označuje v diagramu místo, kde se tok opět spojuje do jednoho toku.

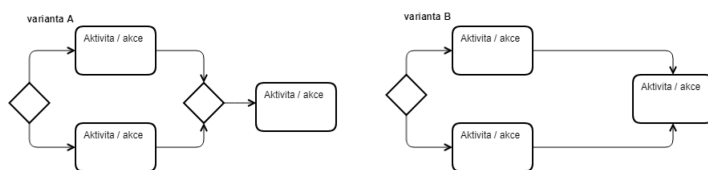
Grafická notace viz obrázek 11 - čtverec otočený o úhel 45 stupňů tzn. stojící na vrcholu, může být připojeno textové označení specifikace uzlu.



Obrázek 11: Uzel sloučení

V rámci použití tohoto uzlu je nutné si uvědomit rozdíl použití toku z tímto uzlem a bez něj jak ukazuje obrázek 12. Pokud je použita varianta A, pak se do následující akce pokračuje ihned, jakmile se do uzlu sloučení dostane minimálně jeden z toku. V případě varianty B je následující aktivita zahájena až ve chvíli, kdy je dosažena oběma toky.

1.2.3.6 Rozvětvení (Fork node) je uzel, který označuje v diagramu místo, kde se tok rozděluje do více větví a pokračuje ve všech větvích paralelním tokem.



Obrázek 12: Příklad rozhodnutí

Grafická notace viz obrázek 13 - vyplněný obdélník, u kterého může být připojeno textové označení specifikace uzlu, může být umístěn v horizontální nebo vertikální poloze v závislosti zobrazení toku.



Obrázek 13: Uzel rozvětvení

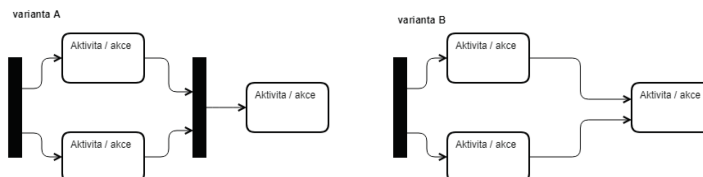
1.2.3.7 Sjednocení (Join node) je uzel, který označuje v diagramu místo, kde se tok opět spojuje do jednoho toku.

Grafická notace viz obrázek 14 - vyplněný obdélník, u kterého může být připojeno textové označení specifikace uzlu, může být umístěn v horizontální nebo vertikální poloze v závislosti zobrazení toku.



Obrázek 14: Uzel sjednocení

V rámci použití uzlu sjednocení je možné využít alternativní zápis a zcela jej vynechat, jak je znázorněno na obrázku 15. Oba dva zápisy mají ekvivalentní výsledek, následná akce je vždy provedena až ve chvíli, kdy jsou obě větve toku vykonány.



Obrázek 15: Příklad paralelního toku

1.3 Vztahy mezi prvky

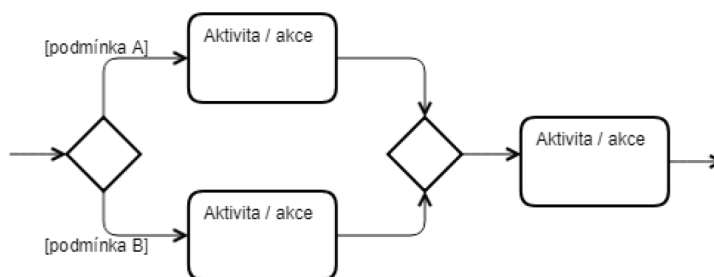
Diagram aktivit je diagramem chování, a proto vztahy mezi uzly, které zachycuje, a především ty dynamické, jsou tím klíčovým přínosem jeho využití.

1.3.1 Hrany

Pro zobrazení dynamických vztahů, které zachycují chování modelovaného systému, jsou používány hrany (Edges), které spojují jednotlivé uzly v diagramu. Výsledkem je zobrazení toků aktivity, a to z několika úhlů. Proto jsou toky v diagramu označovány takto:

1.3.1.1 Řídící tok (Control flow) znázorňuje následnost jednotlivých vykonatelných uzlů, která je řízena pomocí řídicích uzlů, který tento tok usměrňují dle daných pravidel a podmínek.

Grafická notace viz obrázek 16 - plná čára se šipkou na jednom z konců, která vede od jednoho uzlu ke druhému, může být doplněna textovým označením definujícím podmínky toku v hranatých závorkách.

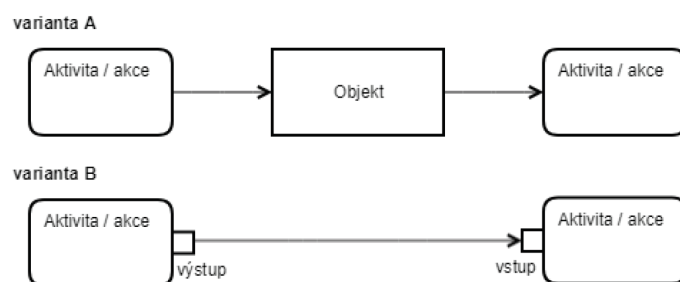


Obrázek 16: Řídící toky

1.3.1.2 Objektový tok (Object flow) znázorňuje následnost vykonatelných uzlů v případě, kdy je potřeba navíc doplnit údaje o fyzickém předávání určitých objektů (ať již ve formě materiálu nebo prostých dat)

Grafická notace viz obrázek 17 - plná čára se šipkou na jednom z konců, která vede od jednoho uzlu ke druhému, přičemž hlavní dvě možná zobrazení jsou prostřednictvím pinů u aktivity nebo vložením objektového uzlu mezi vykonatelné uzly. Notace samozřejmě umožňuje i složitější konstrukty jako více pinů u uzlu aktivity nebo využití řídicích uzlů v případě potřeby zdůraznění paralelního zpracování objektů, v této práci je však zjednodušeno.

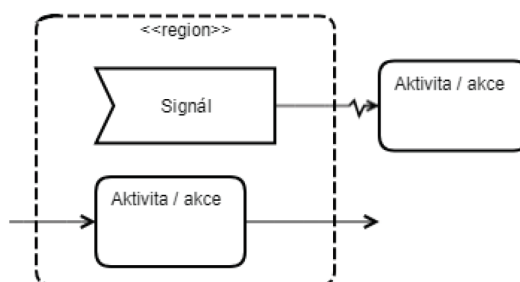
1.3.1.3 Přerušování (Interrupting edge) je specifický příklad hrany, která zobrazuje řídicí tok, a používá se ve spojení s prvkem Regionu, viz obrázek 20, zpravidla k zobrazení zpracování výjimky (například se vyskytne nějaká chyba v rámci



Obrázek 17: Objektové toky

vykonávání nějaké činnosti / akce, která ukončí tuto činnost, zalogue její průběh apod.)

Grafická notace viz obrázek 18 - plná čára se šipkou na jednom z konců, která vede od jednoho uzlu ke druhému, doplnění o miniaturu „blesku“, alternativně je přímo samotná čára ve tvaru [3] „blesku“.



Obrázek 18: Přerušení

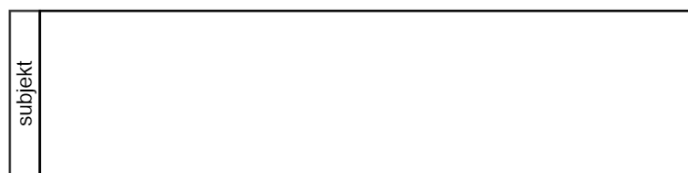
1.3.2 Oddíly

Účelem Oddílů (Partions) je znázornit odpovědnost za vykonávání jednotlivých činností účastníkům procesu / aktivity, tedy jednotlivých subjektům, jimiž mohou být pracovníci, oddělení, systémy. Běžné zobrazení je ve formě tzv. „plavecké dráhy“ (swimlanes) [1].

Grafická notace viz obrázek 19 - obdélník v horizontální nebo vertikální poloze, v jehož „záhlaví“ je uveden název subjektu, který znázorňuje. V diagramu jich může být zobrazeno více a mohou být spojeny do bloků přilehnutím hran k sobě.

1.3.3 Oblasti

Oblast (InterruptibleActivityRegion) je skupina aktivit, která podporuje ukončení části aktivity. Jsou tedy úzce spojeny s uzly akce, se zachytáváním signálu a s použitím tzv. hrany přerušení.



Obrázek 19: Plavecká dráha

Grafická notace viz obrázek 20 - obdélník se zaoblenými rohy s obrysem znázorněným přerušovanou čarou.



Obrázek 20: Oblast - region

2 Zadání a požadavky

Cílem práce je sestavit aplikaci pro vytváření diagramů aktivit, které modelují chování procesů nebo je lze použít pro popis složitějších algoritmů. Aplikace bude určena pro uživatele z řad pracovníků společností, kteří začínají s notací UML a kteří se se pohybují mezi IT a business (tímto jsou myšleno, že jsou z jiných, než IT oddělení, a potřebují tedy jednoduchou a pochopitelnou cestu, respektive nástroj, pro komunikaci s IT pracovníky).

2.1 Příklady programů

Na trhu existuje celá řada aplikací poskytující možnosti tvorby diagramů v UML notaci [4]. Uživatelé si mohou vybrat ze široké nabídky aplikací podle různých parametrů:

- licenční přístupnosti
 - placené aplikace „SPARX Enterprise Architect“
 - neplacené „Modelio“
- technické platformě
 - desktopové okenní aplikace jako „Visual Paradigm“
 - webové aplikace (ať již v licenční verzi cloud či on-premise) jako například „Gliffy“

- šířky použitelnosti
 - specializované výhradně na UML jako „UML Diagram Maker“
 - multinotační jako „Gliffy“

Shodná je však u většiny aplikací komplexnost, protože obsahují více typů diagramů notace UML. Podobná je také většinou ovládací logika, protože aplikaci obsahují po stranách panel s prvky, které se mohou přetažením vkládat do diagramu, v němž se následně mohou pomocí kursoru upravovat.

Zároveň obsahují poměrně široké možnosti pro uživatele, co se týče vkládání prvků, jejich atributizace a použití. Tím pádem se však pro začátečníka v UML jeví na první pohled velmi složité.

2.2 Požadované funkce a vlastnosti aplikace

Na vytvářenou aplikaci jsou dány následující základní požadavky plynoucí z cíle práce a průzkumu stávajících řešení:

1. Uživatelské prostředí aplikace
 - (a) Aplikace bude minimalistická pro jednoduchou orientaci uživatele.
 - (b) Aplikace bude obsahovat panel nástrojů, ve kterém budou jednotlivé prvky diagramu.
 - (c) Prvky diagramu bude možné umísťovat na kreslicí plochu po výběru z panelu nástrojů.
 - (d) Aplikace bude obsahovat základní a nejčastější UML prvky diagramu aktivit.
 - (e) Aplikace bude umožňovat použití klávesových zkratk.
 - (f) Aplikace bude umožňovat funkci Drag&Drop pro práci s jednotlivými prvky diagramu.
 - (g) Aplikace bude obsahovat základní kontrolu UML notace.
 - (h) Aplikace bude nezávislá na připojení do internetu / sítě.
2. Uložení diagramu
 - (a) Vytvořený diagram bude možné uložit do souboru ve formátu XML a následně zpětně načíst do aplikace.
 - (b) Vytvořený diagram bude možné uložit do souboru.
 - (c) Vytvořený diagram bude možné uložit do souboru s názvem dle diagramu.
3. Export diagramu
 - (a) Vytvořený diagram bude možné exportovat do formátu JPEG a PNG.

- (b) Vytvořený diagram bude možné exportovat do zvoleného umístění.
- (c) Vytvořený diagram bude možné exportovat do souboru s názvem dle diagramu nebo jiným uživatelsky zadaným.

3 Technická dokumentace

Následující část dokumentace je věnována popisu aplikace z technického pohledu.

3.1 Technologie

Návrh programu (struktura namespace, tříd, UI) byl prováděn v programu **Gliffy**. Vybrané důležité diagramy jsou součástí této práce.

Samotný vývoj programu byl prováděn v aplikaci **Microsoft Visual Studio**, ve verzi Community Edition, 2017. Implementace byla provedena jako desktopová okenní aplikace pro Windows za pomoci těchto technologií:

- programovací jazyk **C#**
- **WPF** (Windows Presentation Forms) - zvoleno jako výkonnější varianta WinForms vzhledem k práci s grafickými prvky v diagramu
- **MS .NET Framework**, ve verzi 4.6.1
- **Inno Setup**, pro vytvoření instalátoru aplikace (převzato z [6])

Pro ukládání programu (vytvořených diagramů) bylo zvoleno ukládání do **XML** (eXtensible Markup Language), prostřednictvím třídy **NetDataContractSerializer**. Tato třída je součástí DLL **System.Runtime.Serialization**, jež byla do projektu přidána.

3.2 Struktura programu

Projekt, přesněji jednotlivé třídy, jsou rozděleny do pěti namespace podle jejich logické souvislosti. Namespace jsou v projektu prezentovány stejnojmennými adresáři pro lepší orientaci. Struktura aplikace je blíže naznačena v diagramu, který je součástí práce jako příloha č.1.

3.2.1 ActivityDiagram.AppDiagram

AppDiagram sdružuje základní třídy aplikace, které mají za úkol pracovat s diagramem a uživatelským rozhraním:

- **Diagram** - klíčová třída aplikace, třída pro zachycení jednotlivého diagramu, která obsahuje základní nastavení diagramu, kreslicí plátno canvasDiagram, dále TabItem, ve kterém je uložen, a především seznam uzlů a spojnic (jako atributy listGO a listGConnect),

- **MenuOfControl** - třída ovládacího panelu pro volbu módu aplikace či diagramu,
- **ControlsItem** - třída obsahující menuButton a atribut, zda je aktivní nebo není, pro určení módu aplikace či diagramu,
- **UmlRuleChecker** - statická třída, která poskytuje metody pro kontrolu pravidel UML notace (aktuálně pouze požadované základní),
- **Mode** - výčtový typ pro mód, ve kterém pracuje aplikace, respektive diagram; odpovídá názvům jednotlivých menuButton v MenuOfControl, pouze Mode "grab" neodpovídá žádnému tlačítku, ale jejich vyresetování (v tomto módu je možný drag&drop prvků diagramu (přesněji pouze uzlů)).

3.2.2 ActivityDiagram.AppWindows

AppWindows sdružuje třídy a XAML definice jednotlivých oken používaných aplikací:

- **AppMainWindow** - jedná se o hlavní okno aplikace, které zprostředkovává komunikaci uživatele s diagramem (obsahuje ovládací panel a založky s jednotlivými diagramy), obsahuje také Resources pro menuButton z ovládacího panelu.
- **CloseDiagramWindow** - okno volané z AppMainWindow při zavírání diagramu případně ukončování aplikace,
- **HelpWindow** - okno volané z AppMainWindow a obsahující textovou nápovědu pro uživatele,
- **RenameDiagramWindow** - okno volané z AppMainWindow a obsahující funkcionality umožňující uživatelskou změnu názvu diagramu, který byl při vyvolání aktivní v okně AppMainWindow,
- **ResizeForkJoinWindow** - okno je voláno z kontextového menu prvku třídy Grafic, konkrétně GOforkJoinNodeHorizontal a/nebo GOforkJoinNodeVertical, a slouží k nastavení velikosti těchto prvků na plátně (nastavuje enum Size těchto prvků).

3.2.3 ActivityDiagram.GraficClass

GraficClass obsahuje třídu Grafic včetně všech jejích zděděných tříd a jimi používaných enum, struktur a internal nebo private tříd. Více je třída popsána ve stejnojmenné kapitole.

3.2.4 ActivityDiagram.HelperFactory

Sdružuje obecné pomocné třídy pro aplikaci, které jsou statické:

- **ExportImageFactory** - poskytované metody provádějí uložení kreslicího plátna diagramu do souboru se zadaným názvem dle volané metody,
- **XmlTransformFactory** - poskytované metody provádějí uložení a načtení diagramu včetně testu provedení akce,
- **MyGeometry** - poskytované metody provádějí obecné výpočty potřebné pro vykreslování diagramu.

3.2.5 ActivityDiagram.Notification

Třídy v něm obsažené jsou všechny statické a slouží pro zobrazování notifikací z aplikace uživateli:

- **Notify** - obecně je při potřebě volána metoda Show, která přijímá enum TypeAction (zde je nutné rozšiřovat výčet typů akce, z níž je notifikace volána) a boolean (true je označení pro úspěšné vykonání a false pro neúspěšné vykonání akce),
- **TypeAction** - výčet možných akcí či operací v aplikaci, ze kterých je spouštěna notifikační služba,
- **ResultString** - internal třída, která provádí vyhodnocení a samotné provedení notifikační akce.

3.3 Popis třídy Graphics

Graphics je základní třída pro prvky na kreslicím plátně canvasDiagram, hrany i uzly. Jedná se přesněji o celou děděnou strukturu tříd, která je znázorněna na diagramu v příloze č.2. Každý objekt Diagram obsahuje seznam objektů GraphicsObject, které zastupují uzly diagramu, a seznam objektů GraphicConnector, které zastupují hrany diagramu.

V rámci adresáře v projektu (Solution ve VS) jsou třídy organizovány do podadresářů. Podadresáře jsou pojmenovány podle třídy, kterou obsažené třídy dědí. V rámci této stromové struktury jsou třídy označeny jako abstract, s výjimkou koncových tříd. Koncové třídy je možné implementovat jako reálné prvky/objekty diagramu a jsou označeny jako sealed třídy. Další popis je zaměřen na sealed třídy.

3.3.1 GaficsObject

Struktura GraphicsObject je větvena/děděna podle společných charakteristik prvků (umístění prvku textbox, obsahu kontextového menu, možnosti úpravy velikostí). Povinně je nutné v sealed metodách definovat metodu DrawObject, pro vykreslení objektu.

3.3.1.1 GONode

- **GObjectNode** - slouží k zobrazení objektového uzlu v módu diagramu „objectNodeButton“, označené v UI jako „Object“,
- **GOActivityNode** - slouží k zobrazení uzlu akce, alternativně aktivity, v módu diagramu „activityNodeButton“, označené v UI jako „Activity / Action“,
- **GOinitNode** - slouží k zobrazení uzlu počátku diagramu v módu diagramu „initNodeButton“, označené v UI jako „Initail node“,
- **GOactivityFinalNode** - slouží k zobrazení uzlu konce aktivity v módu diagramu „activityFinalNodeButton“, označené v UI jako „Activity final“,
- **GOflowFinalNode** - slouží k zobrazení uzlu konce jednoho toku v rámci aktivity v módu diagramu „flowFinalNodeButton“, označené v UI jako „Flow final“,
- **GOforkJoinNodeHorizontal** - slouží k zobrazení uzlu rozvětvení nebo sjednocení v módu diagramu „forkJoinNodeButtonHorizontal“, označené v UI jako „Fork / join“ horizontální ikona,
- **GOforkJoinNodeVertical** - slouží k zobrazení ... v módu diagramu „forkJoinNodeButtonVertical“, označené v UI jako „Fork / join“ vertikální ikona,
- **GOdecMergeNode** - slouží k zobrazení uzlu rozhodnutí a sloučení v módu diagramu „decMergeNodeButton“, označené v UI jako „Decision / Merge“,
- **GOtime** - slouží k zobrazení uzlu časové události v módu diagramu „timeButton“, označené v UI jako „Time“,
- **GOsend** - slouží k zobrazení uzlu odeslání signálu v módu diagramu „sendButton“, označené v UI jako „Send“,
- **GOrecieve** - slouží k zobrazení uzlu odeslání signálu v módu diagramu „receiveButton“, označené v UI jako „Recieve“,

3.3.1.2 GOArea Třída obsahuje strukturu `ResizeConnors` složenou z objektů tříd `ResizeConner`. Jejich prostřednictvím je umožněno provádět uživateli změnu velikosti uzlu (středově orientované zvětšení objektu uzlu).

- **GOAreasArea** - slouží k zobrazení uzlu označujícím oblast, případně strukturovaný uzel, v módu diagramu „areaButton“, označené v UI jako „Region“,

- **GOAreasNotice** - slouží k zobrazení uzlu pro poznámku v módu diagramu „noticeButton“, označené v UI jako „Notice“,
- **GOAreasPartionVertical** - slouží k zobrazení vertikálně orientované plavecké dráhy v módu diagramu „verticalButton“, označené v UI jako „Vertical“,
- **GOAreasPartionHorizontal** - slouží k zobrazení horizontálně orientované plavecké dráhy v módu diagramu „horizontalButton“, označené v UI jako „Horizontal“.

3.3.2 GfraficConnector

Třída představuje prvek hrany v diagramu. Obsahuje strukturu EndNodes, která odkazuje na uzly (dva objekty GraphicsObject) diagramu, které hrana spojuje. Voláním ResetPosition dochází vždy k překreslení hrany do správné polohy vůči těmto koncovým uzlům. Tato třída je abstract, konkrétně jsou implementovány jako sealed třídy tyto, používané v diagramech:

- **GCControlFlow** - slouží k zobrazení jednoduché hrany mezi uzly diagramu v módu diagramu „controlFlowButton“, označené v UI jako „Standard edge“,
- **GXObjectFlow** - slouží k zobrazení jednoduché hrany mezi uzly diagramu s objektovými piny na koncích v módu diagramu „objectFlowButton“, označené v UI jako „Edge with pins“,
- **GCInterruptFlow** - slouží k zobrazení hrany přerušování mezi uzly diagramu v módu diagramu „interruptFlowButton“, označené v UI jako „Interruptin Edge“.

4 Uživatelská dokumentace

Následující část dokumentace je věnována popisu aplikace z uživatelského pohledu.

4.1 Instalace aplikace

Aplikace byla testována na zařízení následující konfigurace, která je zároveň určena jako minimálně pořadovaná konfigurace pro tuto aplikaci:

- požadavky SW
 - Windows 10 Home (verze 1803)
 - .NET Framework 4.6.1
- požadavky HW

- Intel Core i7-7500U CPU 2.70GHz 2.90 GHz
- SSD 1 TB (volný prostor pro instalaci minimálně 68 MB)
- RAM 8,00 GB

Instalace probíhá pomocí instalátoru (průvodce instalací), který se spustí otevřením souboru „setupActivityDiagram“. Během instalace se provádí nejdříve kontrola potřebného .NET Frameworku ve verzi 4.6.1 v zařízení a jeho případní doinstalace (vše je součástí instalačního balíčku). Následně se složky nakopírují do ProgramFiles adresáře a spustí (na dotaz) instalace samotného programu. Po jejím dokončení je na ploše ikona zástupce ke spuštění aplikace.

Odinstalace pak probíhá sputěním souboru Uninstall ve stejném adresáři a následným odstraněním programu v nabídce Přidat nebo odebrat programy ve Windows.

4.2 Ovládací prvky

4.2.1 Hlavní okno aplikace

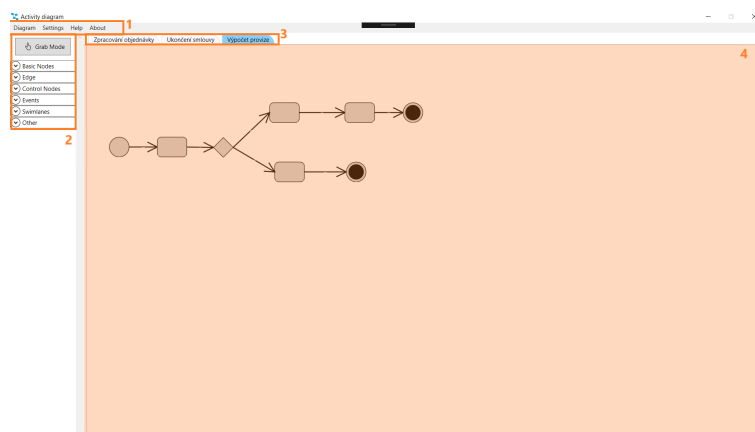
Hlavní okno obsahuje následující prvky pro ovládání aplikace:

1. **Menu aplikace** umožňuje základní uživatelské ovládání aplikace, rozděleno do těchto hlavních voleb:
 - **Diagram** obsahuje volby, které umožňují uživateli přidat nový diagram, otevřít uložený diagram, exportovat aktivní diagram do souborů JPEG nebo PNG, uložit diagramy do souboru.
 - **Settings** obsahuje volbu pro změnu barevného schématu aktivního diagramu.
 - **Help** obsahuje základní uživatelskou nápovědu k aplikaci včetně klávesových zkratk, které jsou v aplikaci implementovány.-
2. **Panel pro vkládání prvků diagram** je blíže popsán dále.
3. **Záložky diagramů** pro přepínání mezi více diagramy otevřenými v aplikaci.
4. **Kreslicí plátno diagramu** pro zobrazení a editaci diagramu.

4.2.2 Panel pro vkládání prvků diagramu

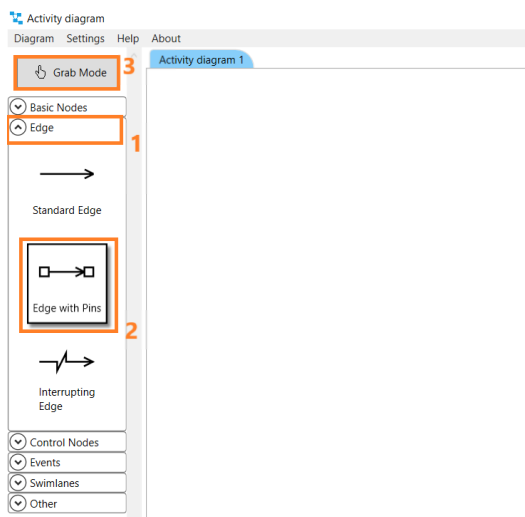
Panel slouží uživateli ke zvolení prvku, který se má do diagramu vložit, ať již uzel nebo hrana mezi dva uzly. Obsahuje tyto ovládací prvky:

1. **tlačítko skupiny prvků** - stiskem otevře příslušnou nabídku tlačítek prvků, a zavře jinou skupinu prvků, pokud je otevřena,



Obrázek 21: Hlavní okno aplikace

2. **tlačítko prvku** - stiskem dojde k aktivaci módu spojených s tímto tlačítkem a uživatel může v diagramu přidávat uzly nebo je spojovat (v závislosti na módu),
3. **tlačítko Grab Mode** - stiskem dojde k nastavení módu, který umožňuje uživateli pomocí způsobu drag&drop jednotlivé uzly na kreslicím plátně diagramu přesouvat.



Obrázek 22: Panel pro vkládání prvků diagramu

4.3 Práce s grafickými prvky diagramu

Diagram, respektive kreslicí plátno, pracuje v módech, které je možné rozdělit na tři typizované skupiny módů podle chování při kliknutí do kreslicího plátna:

- **vložení uzlů** klikem do volného místa (v módu například Region, Time nebo Object),
- **vytvoření hran** kliknutím na jeden uzel - počáteční a druhý uzel - koncový; kliknutím na již označený uzel, dojde k odznačení uzlu a je možné vybrat jiný počáteční uzel (v módu například Standard Edge),
- **přesun uzlů** klikem do vloženého uzlu dojde k jeho označení zeleným rámečkem (aktivní prvek) a je možné s ním pohybovat po plátně metodou drag&drop, případně ovládat aktivní prvek na klávesnici - pohyb prvku stiskem kláves šipek, vymazání prvku stiskem klávesy Del, multivýběr prvků držením klávesy Ctrl (obecně Grab Mode).

Zároveň každý prvek diagramu obsahuje kontextové menu, které se zobrazí umístěním kurzoru na prvek a stiskem pravého tlačítka). Při stisku pravého tlačítka na plátnu mimo jakýkoliv prvek se zobrazí menu patřící danému diagramu.

Závěr

Aplikace vytvořená v rámci této práce splňuje požadavky a zadání specifikované blíže v kapitole 2. Umožňuje tedy uživatelům „UML-začátečnickům“ tvorbu jednoduchých diagramů aktivit a jejich následný export či uložení. Zároveň těmto uživatelům poskytuje stručný úvod do notace UML a uživatelského používání této aplikace pro jejich snazší začátek. V práci je také obsažena technická dokumentace pro případné zájemce o její další rozšiřování.

4.4 Zhodnocení a náměty

Autor zpětně spatřuje určité možnosti optimalizace kódu týkající se lepšího rozdělení úloh mezi třídy (například vyčlenění určité diagramové logiky z `AppMainWindows`), jednodušší implementace některých tříd (například `ControlButton` mohla být rozšířená třída zděděná od třídy `Button`) nebo případně větší granularitu tříd, co se týká pomocných tříd, v rámci třídy `Grafic` (například vyčleněné obrysy do vlastní třídy). A samozřejmě také instalátoru aplikace.

Aplikace je však z funkčního hlediska plnohodnotnou aplikací splňující očekávané požadavky ze zadání a uživatel tyto nedostatky nepocítuje. Autor dále vidí možný rozvoj aplikace dvěma směry:

- intenzivně, tedy rozvíjet aplikaci ve směru zlepšení uživatelského rozhraní a přidání funkcionalit pro práci s implementovaným diagramem aktivit (například zarovnání prvků v diagramu, zalomené hrany diagramu nebo změna barevnosti pouze vybraných prvků v diagramu),
- extenzivně, tedy rozšířením aplikace o další typy diagramů notace UML.

Conclusions

The applications created within this work meet the requirements and specifications specified in Chapter 2. It enables “UML-beginners” to create simple activity diagrams and export or save them. At the same time, this thesis provides a brief introduction to the UML notation and makes easier start to user’s usage of this application. The thesis also contains technical documentation for potential candidates for further enlargement.

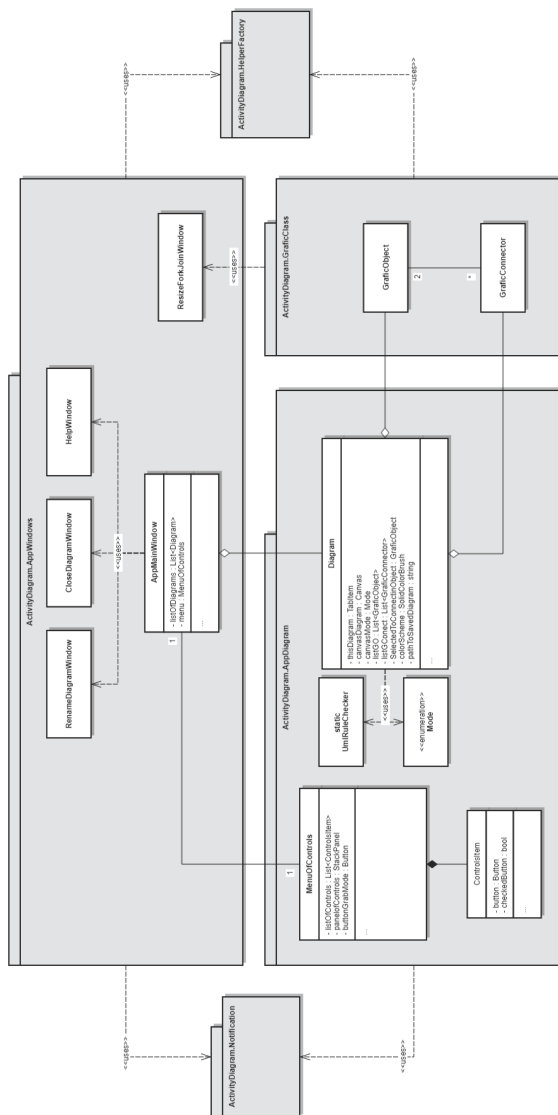
4.5 Assessments and Suggestions

The author is aware of possible optimization of the code in better division of tasks between classes (for example, the allocation of certain diagram logic from AppMainWindows), simpler implementation of some classes (such as Control-Button could be an extended class inherited from the Button class) or larger class granularity as far as auxiliary classes, within the Grade class (for example, outlined contours in a custom class). And of course installer of application.

However, the application is a full-featured application meeting the expected requirements of the assignment, and the user does not feel these shortcomings. The author sees the possible development of the application in two directions:

- intensively, ie. to develop the application in the direction of improvement of the user interface and to add functionalities for working with the implemented activity diagram (for example alignment of elements in the diagram, angled edges of the diagram or change of the color of only selected elements in the diagram),
- extensively, ie. extending the application to other types of UML notation diagrams.

A Příloha č.1



Obrázek 23: Zobrazení struktury programu

C Obsah příloženého CD/DVD

bin/

Instalátor INSTALLACTIVITYDIAGRAM programu spustitelný přímo z CD/DVD, který obsahuje i všechny soubory potřebné pro bezproblémový běh instalátoru z CD/DVD.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové texty programu ACTIVITYDIAGRAM se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnamy a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu.

ReadMe.txt

Instrukce pro instalaci a spuštění programu ACTIVITYDIAGRAM, včetně všech požadavků pro jeho bezproblémový provoz.

Literatura

- [1] CAROL BRITTON AND JILL DOAKE. A student guide to object-oriented development. Amsterdam: Elsevier Butterworth-Heinemann, 2005. ISBN 0750661232.
- [2] Unified Modeling Language (UML) description, UML diagram examples, tutorials and reference for all types of UML diagrams - use case diagrams, class, package, component, composite structure diagrams, deployments, activities, interactions, profiles, etc. [online]. 2009 [cit. 01.08.2018]. Dostupné z: <http://www.uml-diagrams.org>
- [3] Unified Modeling Language (UML) description, UML diagram examples, tutorials and reference for all types of UML diagrams - use case diagrams, class, package, component, composite structure diagrams, deployments, activities, interactions, profiles, etc. [online]. 2009 [cit. 01.08.2018]. Dostupné z: <http://www.uml-diagrams.org/activity-diagrams.html>
- [4] 40 Open Source, Free and Top Unified Modeling Language (UML) Tools in 2018. PAT Research: B2B Reviews, Buying Guides & Best Practices [online]. 2018 Predictive Analytics Today. All rights reserved. [cit. 31.07.2018]. Dostupné z: <https://www.predictiveanalyticstoday.com/open-source-free-unified-modeling-language-uml-tools/>
- [5] Designing Activity Diagrams. Design Patterns & Refactoring [online]. 2007 [cit. 31.07.2018]. Dostupné z: <https://sourcemaking.com/uml/modeling-business-systems/external-view/constructing-activity-diagrams>
- [6] Installing .NET Framework 4.6.1 with Inno Setup, GitHub. [online]. 2018 [cit. 31.07.2018]. Dostupné z: <https://gist.github.com/canton7/72104b1fb154442d5a9ba937fd3ee781>
- [7] UML (Unified Modeling Language). Home [online]. Dostupné z: <http://lucie.zolta.cz/index.php/iformacni-systemy-databaze/33-uml-unified-modeling-language>
- [8] About the Unified Modeling Language Specification Version 2.5.1. OMG | Object Management Group [online]. 2018 [cit. 01.08.2018]. Dostupné z: <https://www.omg.org/spec/UML/2.5.1>