

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra matematiky

## **Bakalářská práce**

# **Implementace algoritmů pro šíření nebezpečných látek**

Plzeň, 2009

Michal Kepka



Rád bych zde poděkoval vedoucímu mé bakalářské práce, panu Ing. Janu Ježkovi, za metodické vedení, věcné připomínky a za podněcení zájmu o open source GIS.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 1. 6. 2009

## **Abstrakt**

Hlavním cílem této práce je implementovat vybraný algoritmus šíření nebezpečných látek podle požadavku Hasičského záchranného sboru ČR. V teoretické části jsou popsány vybrané příručky pro vyhodnocení situace v místě havárie a představena knihovna algoritmů Sextante. V praktické části je podrobně popsán postup implementace a její výsledek v podobě programu vykreslujícího zasažené oblasti.

## **Klíčová slova**

GIS, havárie, nebezpečná látka, Sextante, implementace, algoritmus, ERG2008

## **Abstract**

Dominant theme of this thesis is an implementation of dangerous substances spreading algorithm on request of Fire Rescue Service of Czech republic. At the theoretical part are described two guidebooks which analysis the situation at dangerous substances incident. Below that is open source gealgorithm library Sextante introduced. At the practical part is presented detailed process of implementation and its result in form of affected area visualization.

## **Keywords**

GIS, incident, dangerous goods, Sextante, implementation, algorithm, ERG2008

# Obsah

1 Úvod .....	7
2 Havárie nebezpečné látky .....	8
2.1 Příručka Emergency Response Guidebook 2008 .....	8
2.1.1 Vyhodnocení oblasti havárie .....	9
2.2 Bojový řád jednotek požární ochrany .....	11
2.2.1 Vyhodnocení oblasti havárie .....	11
2.3 Požadavky zadavatele .....	12
3 Implementační prostředí .....	13
3.1 Instalace Sextante .....	13
3.2 Kompilace Sextante .....	13
3.3 Struktura Sextante.....	13
3.4 Programování v Sextante .....	14
3.5 Součásti uživatelského rozhraní Sextante.....	15
3.5.1 Toolbox.....	15
3.5.2 Graphical modeler .....	16
3.5.3 Command-line .....	17
3.5.4 History manager.....	17
3.6 Subjektivní hodnocení Sextante .....	17
4 Postup implementace algoritmu .....	18
4.1 Geometrické vlastnosti obrazce .....	18
4.2 Zadání vektoru větru .....	19
4.3 Zadání místa události .....	20
4.4 Zadání velikosti obrazce .....	20
4.5 Postup výpočtu významných bodů obrazce.....	21
4.6 Uložení a načítání tabelovaných hodnot z příručky ERG 2008 .....	22
4.7 Souřadnicový systém .....	23
4.8 Struktura vytvořeného programu .....	23
4.10 Vstupní okno uživatelského rozhraní .....	24
4.11 Výstupní polygony.....	25
4.12 Shrnutí implementace .....	26
5 Závěr .....	28
Literatura.....	29
Obsah přiloženého CD.....	30

# 1 Úvod

S rozvojem informačních technologií souvisí i rozvoj specializovaných informačních systémů, jakými jsou právě geografické informační systémy (GIS), které dnes již nejsou využívány pouze osobami a organizacemi úzce specializovanými na práci s geoinformacemi. GIS jsou dnes podstatně více využívány širší veřejností, i díky relativně snadnější dostupnosti digitálních geodat.

Důvodem vzniku této práce je právě nahrazení původních postupů při práci s mapovými podklady prací v GIS. Původně se evakuační a ochranné zóny do mapových podkladů zakreslovaly ručně, později v grafických programech, což ovšem bylo jednak pracné a jednak vznikl statický obrázek, se kterým nebylo možno nijak dále pracovat.

Proto bylo Oddělení geomatiky kontaktováno Odborem ochrany obyvatelstva a krizového řízení Generálního ředitelství Hasičského záchranného sboru České republiky s cílem implementovat do prostředí GIS algoritmus vyhodnocující místo havárie, při které unikla nebezpečná látka.

Cílem této práce je provést rešerši literatury zabývající se algoritmy pro šíření nebezpečných látek a podle požadavků HZS ČR implementovat zvolený algoritmus do vybrané stávající GIS knihovny.

Teoretická část této práce pojednává o některých dostupných příručkách pro vyhodnocení havárie a také o implementačním prostředí použitém v praktické části.

Praktická část podrobně popisuje postup implementace algoritmu do zvolené GIS knihovny.

## 2 Havárie nebezpečné látky

Havárie je událost, kdy se uchovávaná látka dostane mimo kontrolu a nastane únik mimo kontrolovanou oblast. Látka, která je škodlivá pro člověka a životní prostředí, je označována jako nebezpečná (viz [1] i [2]). V případě havárie nebezpečné látky je proto nutné, co nejrychleji zareagovat a minimalizovat tak možnost ohrožení člověka a životního prostředí. Klíčový poznatek pro zavedení ochranných opatření je odhad velikosti území, kde hrozí přímé ohrožení života, a území, kde je nutno zavést další bezpečnostní opatření k ochraně obyvatel. Důležitým faktorem pro orientaci bezpečnostních zón je v obou dále uvedených příručkách směr větru v místě havárie.

Na přítomnost nebezpečné látky v místě havárie upozorňují výstražné tabulky na přepravním zařízení nebo na skladištní nádobě [1]. Důležitým informačním prvkem je v tomto případě oranžová cedule (viz obr. 2.1), která informuje o přechovávané látce. Tabulka je rozdělena na horní a dolní polovinu, horní obsahuje třídu nebezpečnosti (Kemlerův kód), dolní polovina obsahuje charakteristické čtyřčíslí (UN kód), toto číslo je přiřazeno látkám a jejich směsím v normách OSN a slouží k jejich identifikaci (viz [1], [12]).



Obr.2.1: Identifikační tabulka , zdroj [1]

Následují způsoby vyhodnocení zón ve dvou různých veřejnosti přístupných příručkách pro dané krizové situace. Existují přesnější způsoby vyhodnocení, ale ty jsou již utajovány. V obou uvedených případech se jedná o hrubý odhad území, jehož zpřesnění závisí na mnoha faktorech a dané zpřesnění by měla učinit pouze kvalifikovaná osoba. V prvním případě (kap. 2.1) se jedná o příručku využitelnou i relativně laickou osobou, druhý případ (kap. 2.2) je určen pro vycvičené požární jednotky.

### 2.1 Příručka Emergency Response Guidebook 2008

Emergency Response Guidebook [1] je podle oficiálního podtitulku „Příručka pro osoby reagující na nehodu s výskytem nebezpečných látek v počáteční fázi od incidentu“. Tato příručka byla vyvíjena společně ministerstvy dopravy v Kanadě, Spojených státech, Mexiku a Argentině. Prvotně je určena pro hasiče, policii a další záchranné jednotky, kterým poskytuje návod pro rychlou identifikaci nebezpečné látky a shrnuje rady pro ochranu nejen veřejnosti ale i vlastní osoby, nemůže ovšem zastoupit záchranný výcvik nebo zkušený odborný úsudek. Původně je vytvářena pro nehody na silnicích a železnicích, což je potřeba brát v úvahu při posuzování havárií na stabilních objektech. Příručka obsahuje seznamy nebezpečných látek z norem OSN i jiných mezinárodních předpisů např. ADR, což je Evropská dohoda o mezinárodní silniční přepravě nebezpečných věcí, a RID, což je dohoda o přepravě nebezpečných věcí po železnici. Tato příručka označuje UN kód jako ID No.

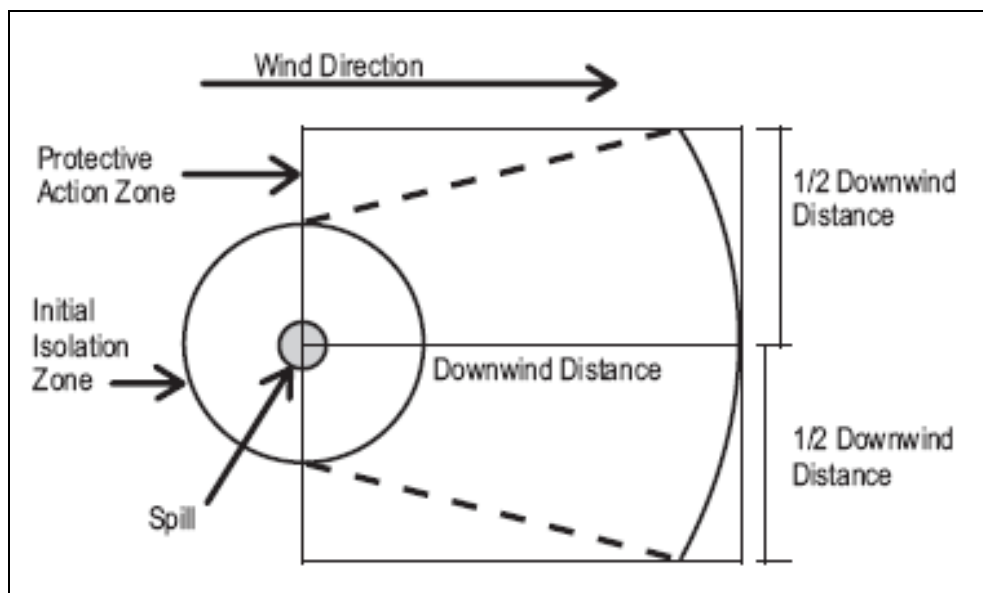


Příručka má čtyři části:

- *žluté stránky* – slouží k vyhledání názvu látky a čísla v průvodci (Guide No.) podle ID čísla (UN kódu)
- *modré stránky* – slouží k vyhledání čísla v průvodci (Guide No.) a ID podle názvu látky
- *oranžové stránky* – tvoří jádro příručky, jsou zde seřazeny skupiny látek (podle Guide No.) a pro každou skupinu jsou sepsána doporučená bezpečnostní opatření v případě požáru, úniku nebo rozlití látky a následná první pomoc. Látky jsou seskupeny podle podobných chemických a toxikologických vlastností.
- *zelené stránky* – obsahují algoritmus pro tvorbu ochranných zón spolu s tabelovanými hodnotami jejich velikostí, v tabulce jsou podle ID čísla seřazeny látky, které vytvářejí toxické výpary a/nebo toxické plyny při kontaktu s vodou, nebo které jsou využívány jako bojové. Mým úkolem je implementovat právě tento algoritmus.

### 2.1.1 Vyhodnocení oblasti havárie

Příručka rozděluje zasažené území do dvou oblastí podle následujícího obrázku 2.2, tento obrazec je označován jako „klíčová dírka“. Dané vzdálenosti, které toto území charakterizují, jsou platné během prvních 30 minut po havárii a jsou uvedeny v tabulce Table 1 – Initial Isolation and Protective Action Distance (viz [1]), v dalším textu jako Table 1.



Obr. 2.2: Obrazec „klíčová dírka“ nad místem havárie, zdroj [1]

*Initial Isolation Zone* („Oblast prvotní izolace“) je území nejbližšího okolí havárie (místo havárie označeno „*Spill*“) o poloměru označovaném v Table 1 jako *First Isolate*. V tomto území mohou být osoby vystaveny nebezpečným (ve směru proti větru) a život ohrožujícím (ve směru po větru) koncentracím látek [1]. V dalším textu budu pro tuto zónu používat český ekvivalent „nebezpečná zóna“ uvedený v Bojovém řádu (viz kap. 2.2).

*Protective Action Zone* („Zóna ochranných opatření“) je území po směru větru od místa havárie, ve kterém mohou být osoby nezpůsobilé a neschopné zavést ochranná opatření a vystavit se tak vážným nebo nevratným zdravotním následkům [1]. Rozměr zóny, označen jako *Downwind Distance*, je uveden v Table 1. Tento rozměr se může podle [1] měnit v závislosti na mnoha faktorech. Například, je-li oblak výparů nebezpečné látky soustředěn v údolí nebo mezi vysoké budovy, vzdálenost narůstá z důvodu nižšího mísení výparů se vzduchem. Vzdálenosti je nutné prodloužit v případě inverze, sněhové pokrývky nebo teploty vyšší než 30°C. Stejně tak, unikne-li s vodou reagující látka do řeky, přesouvá se zdroj výparů s proudem. V nejhrošších případech jako je okamžitý únik veškerého množství látky např. při teroristickém útoku, sabotáži nebo katastrofické havárii, je nutné uvedené vzdálenosti zdvojnásobit. V dalším textu budu pro tuto zónu používat český ekvivalent „zóna ohrožení“ uvedený v Bojovém řádu (viz kap. 2.2).

Vzdálenosti uvedené v tabulce Table 1 jsou podle [1] odvozené z historických dat o nehodách a ze statistického modelu, kdy bylo využito meteorologických pozorování ze 120 míst v USA, Kanadě a Mexiku. Vzdálenosti jsou uvedeny pro malý (do 200 litrů) či velký (přes 200 litrů) únik v noci či ve dne. Pouze pro bojové látky je jako malý označen únik od 2 do 25 kg látky a velký při úniku 25 a více kg látky. Rozdělení hodnot pro den a noc je z důvodu rozdílných atmosférických podmínek v těchto denních dobách. Během dne je atmosféra aktivnější a dochází k lepšímu rozptylování látky, čímž rychle klesá koncentrace a zóna s ohrožující koncentrací je proto menší. Zatímco v noci je atmosféra klidnější, čímž dochází k pomalejšímu rozptylování a území s vysokou koncentrací je větší.

Ochranná opatření zaváděná v jednotlivých zónách musí vést k ochraně zdraví a bezpečí osob [1]. Nebezpečnou zónu je nutné okamžitě opustit ve směru proti větru. V zóně ohrožení se osoby buďto evakuují, je-li k tomuto dostatek času a prostředků, nebo se ukryjí v budovách, je-li evakuace příliš riziková.

K vyhledání údajů v tabulce Table 1 (viz obr 2.3) je nutné znát ID číslo látky (UN kód), známe-li název je nutné nejdříve v modrých stránkách najít ID číslo. Dále je nutné rozhodnout, zda se únik stal ve dne, nebo v noci a zda šlo o malý či velký únik. Není-li jisté o kterou látku se přesně jedná v rámci jednoho ID čísla je nutné použít největší hodnoty vzdáleností pro dané ID číslo [1].

ID No.	NAME OF MATERIAL	SMALL SPILLS (From a small package or small leak from a large package)				LARGE SPILLS (From a large package or from many small packages)			
		First ISOLATE in all Directions Meters (Feet)		Then PROTECT persons Downwind during-		First ISOLATE in all Directions Meters (Feet)		Then PROTECT persons Downwind during-	
				DAY Kilometers (Miles)	NIGHT Kilometers (Miles)			DAY Kilometers (Miles)	NIGHT Kilometers (Miles)
1005	Ammonia, anhydrous	30 m (100 ft)	0.1 km (0.1 mi)	0.2 km (0.1 mi)	150 m (500 ft)	0.8 km (0.5 mi)	2.3 km (1.4 mi)		
1005	Anhydrous ammonia								
1008	Boron trifluoride	30 m (100 ft)	0.1 km (0.1 mi)	0.6 km (0.4 mi)	300 m (1000 ft)	1.9 km (1.2 mi)	4.8 km (3.0 mi)		
1008	Boron trifluoride, compressed								
1016	Carbon monoxide	30 m (100 ft)	0.1 km (0.1 mi)	0.1 km (0.1 mi)	150 m (500 ft)	0.7 km (0.5 mi)	2.7 km (1.7 mi)		
1016	Carbon monoxide, compressed								
1017	Chlorine	60 m (200 ft)	0.4 km (0.3 mi)	1.6 km (1.0 mi)	600 m (2000 ft)	3.5 km (2.2 mi)	8.0 km (5.0 mi)		

Obr. 2.3: Ukázka tabulky Table 1, zdroj [1]

Z tabulky je patrné, že například při malém úniku chlóru ve dne (ID 1017 – Chlorine) je nutné vytvořit nebezpečnou zónu o průměru 60 m a zónu ohrožení o straně 400 m.

## 2.2 Bojový řád jednotek požární ochrany

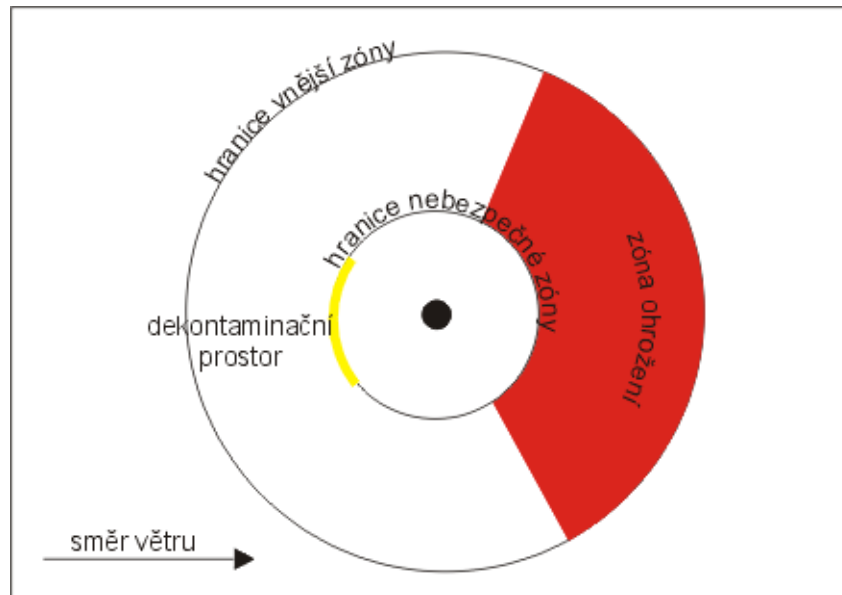
Tato příručka je metodickým návodem pro jednotky požární ochrany, předpokládá tedy výcvik zasahujících jednotek. Proto jsou uvedené hodnoty velice zhruba odhadnuty a jakékoliv zpřesnění je podmíněno rozhodnutím k tomu kvalifikované osoby.

Zásahu s přítomností nebezpečných látek se věnuje kapitola L, která obsahuje jednotlivé metodické listy popisující obecně zásah s přítomností nebezpečné látky, činnost hasičů v jednotlivých zónách, postupy dekontaminace, postup při ropných haváriích, při výskytu výbušných látek a podrobněji zásahy s únikem čpavku a chlóru.

Nebezpečné látky a přípravky jsou ve [2] definovány jako látky a přípravky, které vykazují jednu nebo více nebezpečných vlastností a pro tyto vlastnosti jsou klasifikovány zvláštním zákonem (356/2003 Sb.). Zákon klasifikuje látky jako výbušné, oxidující, extrémně hořlavé, vysoce hořlavé, hořlavé, vysoce toxické, toxické, zdraví škodlivé, žíravé, dráždivé, senzibilující, karcinogenní, mutagenní, toxické pro reprodukci, nebezpečné pro životní prostředí. Bojový řád považuje za havárii nebezpečné látky takovou událost, kdy se nebezpečná látka ocitla mimo kontrolu v tak velkých množstvích, že jsou ohroženi lidé, zvířata a životní prostředí, a je nutné provádět záchranné a likvidační práce.

### 2.2.1 Vyhodnocení oblasti havárie

Není-li zjištěno o jakou látku se jedná, je prvotně ve [2] doporučován odstup 100 m od místa havárie s ohledem na směr větru. Následně se určuje nebezpečná a vnější zóna. Situace rozložení v místě zón ukazuje následující obrázek 2.4:



Obr. 2.4: Zóny podle Bojového řádu

*Nebezpečná zóna* je ve [2] definována jako prostor bezprostředního ohrožení života a zdraví účinky mimořádné události. Tento prostor je ohraničen hranicí nebezpečné zóny a v terénu je rozpoznatelně vyznačen např. páskou. Vzdálenost hranice od nebezpečné látky je závislá na druhu látky. Předběžné hodnoty udává následující tabulka 2.1:

Tabulka 2.1 – poloměr nebezpečné zóny

Látka	Vzdálenost
Hořlavé kapaliny, louhy, kyseliny	5 m
Jedovaté, žíravé plyny a páry	15 m
Látky schopné výbuchu (páry, plyny, prachy)	30 m
Radioaktivní látky	50 m
Výbušniny, rozsáhlá oblaka par	100 – 1000 m

Tyto vzdálenosti jsou doporučené a podle dalších faktorů se mohou měnit např. podle množství látek, možnosti šíření, povětrnostních podmínek nebo konfigurace terénu.

*Zóna ohrožení* je prostor předpokládaného šíření mimořádné události s důsledky na obyvatele nebo objekty.

*Vnější zóna* je podle [3] prostor vymezený pro vedení zásahu, je zde omezen volný pohyb osob a dopravních prostředků, je vymezen hranicí vnější zóny. Minimální dílčí součásti této zóny jsou týlový, nástupní a dekontaminační prostor. Týlový prostor je prostor pro provedení týlových činností (např. soustředění záložních jednotek, odpočinek a stravování, dálková doprava vody) může být umístěn i mimo vnější zónu. Nástupní prostor slouží k soustředění sil a prostředků před nasazením. Dekontaminační prostor se nachází na hranici nebezpečné zóny a slouží k dekontaminaci jednotek a techniky.

V [3] jsou ještě definovány další stanoviště a zóny pro velení, shromažďování osob, poskytování zdravotní péče atd.

## 2.3 Požadavky zadavatele

Na základě komunikace Oddělení geomatiky s GŘ HZS ČR byly formulovány požadavky na implementaci algoritmu uvedeného v ERG2008 ([1], viz kap. 2.1).

HZS definovalo následující požadavky:

- vstupními parametry bude druh látky, místo nehody, rozsah havárie, směr větru
- místo nehody zadávat přenesením souřadnic z podkladové mapy
- výstupem bude vektorová vrstva se dvěma polygony, reprezentující zóny obrazce „klíčová dírka“
- v atributové tabulce obou polygonů uvést druh látky
- uchovávat tabelované hodnoty z ERG2008 v textovém souboru
- umožnit export vytvořené vrstvy do formátu shapefile
- používat souřadnicový systém UTM, případně S-JTSK

## 3 Implementační prostředí

Jako implementační prostředí jsem si vybral knihovnu algoritmů prostorových analýz Sextante (složenina ze „Sistema EXTremeño de ANálisis TErritorial“). Tato knihovna je podle [4] vyvíjena od roku 2004 jako open source platforma pro implementaci a další využívání geoalgoritmů. Knihovna je založena na programovacím jazyce Java společnosti Sun Microsystems, Inc. Z původní nadstavby nad německý systém SAGA se ze Sextante postupně vyvinula nezávislá knihovna algoritmů použitelná jako rozšiřovací modul běžně dostupných GIS programů, v současnosti gvSIG, OpenJump, uDig, OrbisGIS, případně do programu Talent/SDI nebo 52North, který se zabývá Web Processing Service (WPS) službami. Knihovna je svobodný software distribuovaný pod GPL (General Public Licence) licencí (viz [4] a [12]).

### 3.1 Instalace Sextante

V současné době je aktuální verze 0.15, která je dostupná ze stránek projektu [4]. Na stránkách je možno stáhnout zip archiv s jednotlivými jar archivy algoritmů, případně samoinstalační jar archiv připravený přímo pro námi používaný GIS, v současnosti gvSIG a OpenJump. Samozřejmě máme možnost si přes SVN repositář stáhnout samotné zdrojové kódy algoritmů, které můžeme potřebovat při vlastní implementaci nového algoritmu. Na stránkách projektu je také k dispozici dokumentace k algoritmům javadoc a help, ten ovšem pouze ve španělštině.

Instalace spočívá v nakopírování archivů s algoritmy a archivu s třídami „bindings“ (viz dále) do podadresáře uchovávajícího rozšíření dané GIS aplikace. Po tomto kroku můžeme začít používat implementované knihovní algoritmy jako součást GIS aplikace.

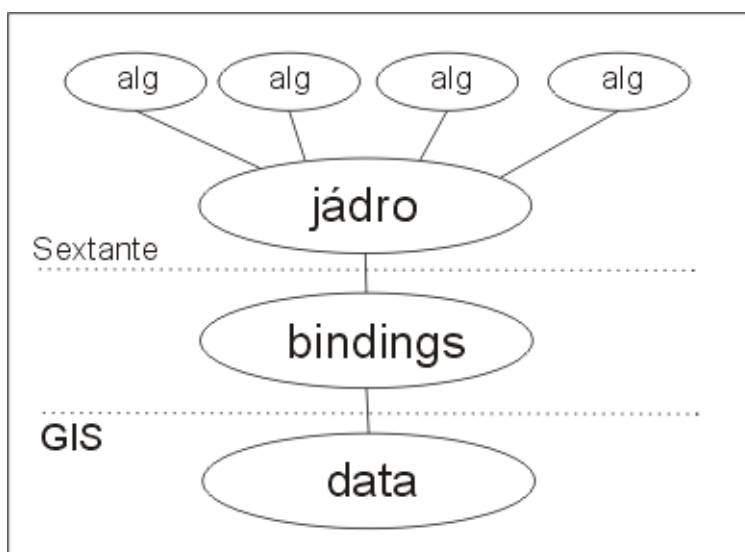
### 3.2 Kompilace Sextante

Ke kompilaci potřebujeme zdrojové kódy stažené z repositáře (URL adresa <https://svn.forge.osor.eu/svn/sextante>), kompilační program, nejlépe Apache Ant, který získáme ze stránek <http://ant.apache.org/> a vývojové prostředí např. Eclipse. V adresáři, kam si z repositáře stáhneme zdrojové soubory, se vytvoří podadresáře s jednotlivými skupinami algoritmů, každá skupina má sestavovací nastavení v souboru *build.xml*. Tento soubor slouží ke kompilaci do jar archivu pomocí Ant a následně lze jar archiv použít jako referenční knihovnu k vlastnímu algoritmu. Naimportujeme-li adresářovou strukturu jednotlivého algoritmu do vývojového prostředí jako Java projekt, můžeme se pustit do úpravy samotného algoritmu, díky licenci GPL.

### 3.3 Struktura Sextante

Před vlastním programováním je vhodné popsat architekturu samotného Sextante. Základem je podle [6] skupina tříd tvořící jádro programu, které definují aplikační programátorské rozhraní (Application Programming Interface – API). Od základních tříd se odvozují samotné geoalgoritmy. Propojení s potřebnými daty z dané GIS aplikace zajišťují tzv. bindings třídy. Které se musí vytvořit pro každou aplikaci, do které chceme Sextante přidat. Ovšem zároveň jsou bindings to jediné, co je nutné ze Sextante měnit,

samotné jádro a algoritmy zůstávají stále stejné. Díky tomu, lze Sextante přidávat do velkého množství GIS aplikací. Strukturu názorně ukazuje následující obrázek 3.1:



Obr. 3.1: Struktura Sextante

Základní součásti, bez kterých není možné algoritmy provádět, se nacházejí v následujících balících:

- *sextante-x.x.jar* – balík s třídami tvořící jádro Sextante (v současnosti *sextante-0.15.jar*)
- *jts-1.9.jar* – balík knihovny Java Topology Suite, který umožňuje ovládat vektorovou geometrii
- *jcommons-1.0.0.jar* – základní součásti knihovny JFreeChart
- *jfreechart-1.0.1.jar* – knihovna JFreeChart, generující tabulky jako výstup algoritmů
- *kxml2.jar* – knihovna XML

Grafické a další součásti nalezneme v knihovnách:

- *sextante\_gui-x.x.jar* – balík s třídami vytvářející grafické uživatelské rozhraní Sextante (v současnosti *sextante\_gui-0.15.jar*)
- *jgraph.jar* – balík vytvářející rozhraní pro tvorbu modelů
- *bsh-2.0b4.jar* – balík obsahující Bean Shell, který je použit pro rozhraní příkazové řádky

### 3.4 Programování v Sextante

Implementace nového algoritmu spočívá podle [6] ve vytvoření projektu, který bude dodržovat strukturu a konvence používané v Sextante. Základem je založit v projektu balík, který dodržuje adresářovou strukturu *es.unex.sextante.názevBalíku*, v mém případě tedy struktura vypadá následovně: *cz.zcu.sextante.ERG*. Je doporučeno pro každý algoritmus zakládat nový balík, který tak obsahuje jen třídy nutné pro vykonání daného

algoritmu. Aby nový algoritmus mohl používat všechny součásti Sextante, je také nutné přidat do referenčních knihoven následující balíky:

- *sextante-x.x.jar* – balík s třídami tvořící jádro Sextante (v současnosti *sextante-0.15.jar*)
- *jts-1.9.jar* – balík knihovny Java Topology Suite (JTS), který umožňuje ovládat vektorovou geometrii
- *jfreechart-1.0.1.jar* – knihovna JFreeChart, generující tabulky jako výstup algoritmů
- *kxml2.jar* – knihovna XML

Nyní můžeme v balíku založit novou třídu s naším algoritmem, zde platí konvence pro název hlavní třídy algoritmu a to taková, že název musí končit příponou *Algorithm*. Každý nově vytvářený geosalgoritmus rozšiřuje základní třídu *GeoAlgorithm*, takže samotná implementace je v podstatě implementací dvou základních metod této třídy.

Metoda *defineCharacteristics* slouží k definování vstupních a výstupních parametrů a zařazení algoritmu do struktury Toolbox (viz dále). Podle této metody se vytváří struktura uživatelského rozhraní při spouštění programu.

V metodě *processAlgorithm* implementujeme samotný algoritmus, zpracováváme vstupní parametry a přiřazujeme hodnoty výstupním.

Máme-li naprogramován vlastní algoritmus, stačí nyní exportovat balík s algoritmy jako jar archiv. Archiv má opět ustálenou konvenci pro název, tedy *sextante\_názevBalíku.jar*. Tento archiv již stačí nakopírovat do adresářové struktury GIS software, kde jsou uloženy ostatní algoritmy Sextante. Při spuštění GIS se načte knihovna Toolbox a námi naprogramovaný algoritmus se objeví v námi zvolené skupině.

## 3.5 Součásti uživatelského rozhraní Sextante

Vložíme-li knihovnu Sextante do podporované GIS aplikace, přidá se na lištu nástrojů nebo jako skupina v menu programu seznam položek uživatelského rozhraní Sextante – Toolbox, Graphical modeler, Command-line, History manager. Následující odstavce jsou zpracovány podle [5].

### 3.5.1 Toolbox

Toolbox (obr. 3.2) je nejpodstatnější část rozhraní, kterou bude uživatel používat nejčastěji. Jedná se o seznam všech implementovaných algoritmů seskupených podle způsobu využití, např. skupina algoritmů pro rastrové nebo vektorové vrstvy, pro hydrologické analýzy atp. v současnosti obsahuje knihovna cca 220 algoritmů, přesné číslo se nachází pokaždé v horní liště okna Toolbox (viz obr. 3.2).

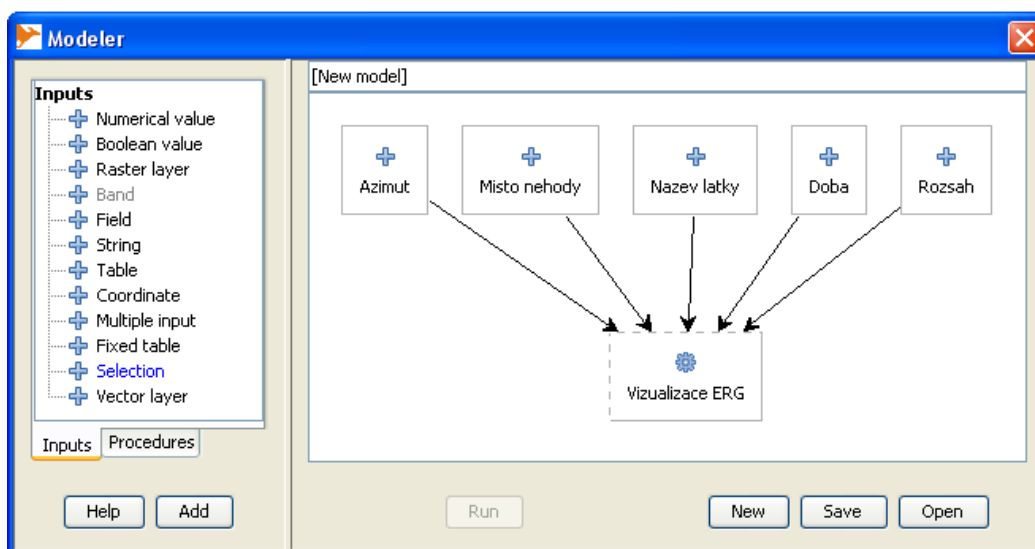
Je-li algoritmus dostupný i bez otevření vrstev ať už rastrových nebo vektorových, je černě podbarven a po dvojkliku se otevře dialogové okno pro zadávání parametrů, není-li bez vrstev dostupný, je název algoritmu šedý a na dvojklik nereaguje. Spustíme-li algoritmus, otevře se dialogové okno, které umožňuje uživateli zadat vstupní hodnoty a další potřebné parametry (viz obr. 4.5). Ať výběrem z rolovací nabídky, vepsáním textového řetězce nebo číselné hodnoty, zadáním cesty k adresáři nebo souboru apod. Stejně tak nabízí možnost zadání názvu a formátu uložení výstupních vrstev.



Obr. 3.2: Toolbox

### 3.5.2 Graphical modeler

Tato neobvyklá součást umožňuje přes jednoduché grafické rozhraní (viz obr. 3.3) vytvářet složité modely analýz ze zřetěžených geoalgoritmů. Čímž odpadá nutnost spouštět každý algoritmus zvlášť a zvlášť nastavovat veškeré parametry. Na obrazovce poskládáme algoritmy do požadovaného pořadí, zadáme vstupní parametry a model proběhne jako jednotlivý algoritmus. Tento modeler by se dobře uplatnil při dalším používání mnou implementovaného algoritmu. Bylo by například možné zřetěžit vizualizaci havárie, průnik výsledného obrazce s vrstvou reprezentující oblasti s rizikovou koncentrací obyvatel, rizikové průmyslové provozy apod. Čímž bychom získali na výstupu nejen vizualizaci zasažené oblasti, ale např. seznam škol, úřadů, podniků v zasažené oblasti.



Obr. 3.3: Modeler



### **3.5.3 Command-line**

V tomto případě se jedná o obdobu klasické příkazové řádky, které je založena na bázi BeanShell. Umožňuje vytvářet pokročilejším uživatelům skripty, které provedou operace rychleji než používáním standardního uživatelského rozhraní, nebo je původní rozhraní ani neumožňuje.

### **3.5.4 History manager**

Jedná se o součást uchovávající historii o provedených příkazech a hodnotách použitých parametrů s časem jejich použití. Schéma uchovávání je stejné, jako kdyby procesy byly prováděny přes Command-line, což je výhodné jednak díky přehlednosti a jednak má uživatel možnost všimnout si struktury používané pro příkazovou řádku a později ji i sám využívat. Uložený proces lze pouhým poklikáním nechat znovu proběhnout.

## **3.6 Subjektivní hodnocení Sextante**

Zpočátku může být pro mnohé uživatele, kteří chtějí implementovat vlastní algoritmus, obtížné proniknout do struktury a konvencí Sextante. Pro mě to také byla ta největší překážka. Při bližším poznání struktury uživatel zjistí, že mu použití Sextante ušetří spoustu další programovací práce. Například má ušetřeno vytváření uživatelského rozhraní, řešení načítání vstupních dat a vytváření struktury výstupních. Tento fakt považuji za velice pozitivní, stejně jako možnost přenositelnosti Sextante do jiných GIS programů a tím i vlastního algoritmu. Na druhou stranu jako nevýhodu považuji nutnost disciplinovaného uživatele při zadávání vstupních dat. Jakákoliv chyba se totiž projeví až při výpočtu algoritmu, ten selže a ukončí se celý proces. Jako slabinu bych také uvedl nedořešenost používání souřadnicových systémů, aby bylo možno nějaká data referencovat musí se k Sextante připojovat jiné knihovny s takovou funkcí, např. knihovna GeoTools. Protože se v současné době na vývoji Sextante živě pracuje, budou nejspíše tyto nedostatky brzy odstraněny. Ze Sextante se tak může stát velice vhodný nástroj jednak pro rychlou implementaci jednoduchých algoritmů, jednak i pro analýzu složitějších problémů.

## 4 Postup implementace algoritmu

Průběh zpracování úkolu lze v podstatě shrnout do čtyř kroků, které ovšem nebyly přesně ohraničeny a v průběhu zpracovávání se činnosti překrývaly a jednotlivé podbody se měnily nebo upravovaly.

Prvním krokem při zpracování bylo zkoumání vlastností požadovaného obrazce a hledání nejvhodnějšího postupu pro algoritmizovanou konstrukci. Především jsem se snažil nalézt jednoduchý postup, který by bylo možno snadno naprogramovat a přitom zachovat požadované vlastnosti obrazce.

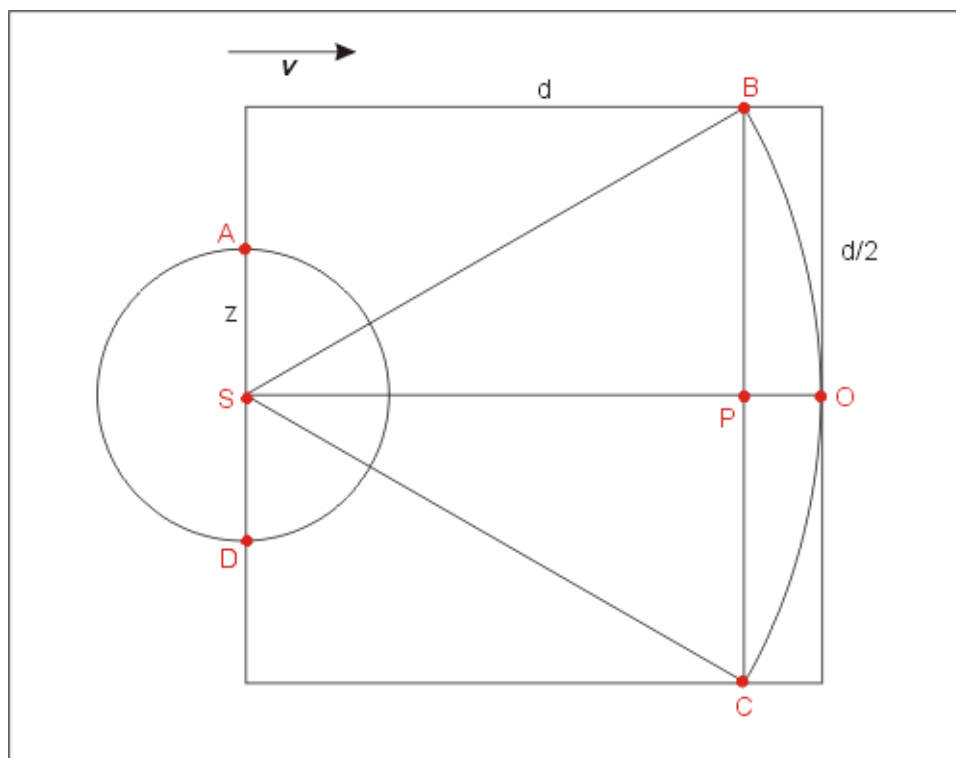
Druhým krokem bylo nalezení nejvhodnějšího způsobu zadávání potřebných parametrů, zvláště důležité bylo nalézt vhodný způsob zadávání směru větru a dále pak způsob uchování tabelovaných hodnot a jejich načítání.

Třetím krokem byla algoritmizace výpočtu bodů obrazce a jejich uložení do struktury tvořící polygony.

Posledním krokem bylo celkové sestavení programu, jeho vložení do zvoleného implementačního prostředí a následné testování správné funkčnosti.

### 4.1 Geometrické vlastnosti obrazce

Na obrázku 4.1 je náčrt základu obrazce s pomocnými konstrukcemi, na kterém budu popisovat vlastnosti, které používám při samotné konstrukci.

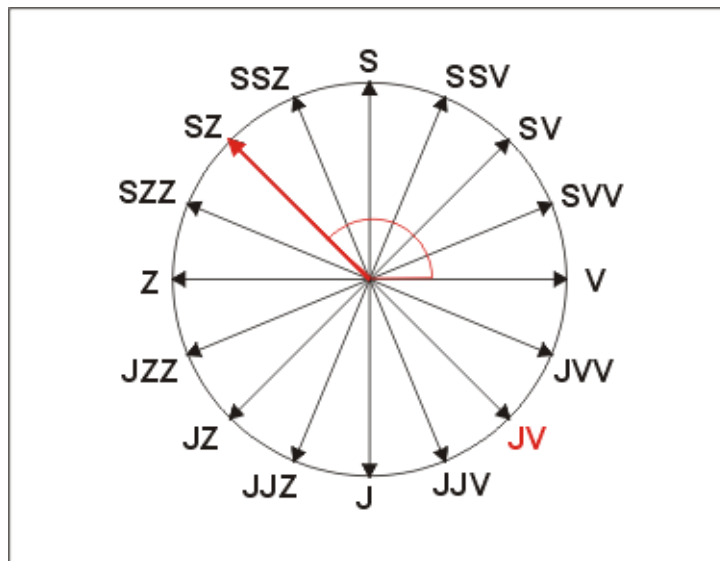


Obr. 4.1: Náčrt obrazce s vyznačením bodů

1. Obrazec se skládá z kružnice se středem  $S$  a poloměrem  $z$ , dále z kruhové výseče se středem  $S$  a poloměrem  $d$ . Výseč je omezena na vnitřní prostor čtverce o straně  $d$ , který se dotýká bodu  $S$  v polovině délky jedné strany.
2. Z těchto předpokladů tedy platí, že vzdálenost úseček  $SA$  a  $SD$  je rovna  $z$ , dále, že úsečka  $SO$  je kolmá na úsečku  $AD$  a její velikost je rovna  $d$ , tedy  $d$  je poloměr výseče. Označíme-li krajní průsečíky výseče a čtverce jako body  $B$  a  $C$ . Vytvoří body  $SBC$  rovnostranný trojúhelník se stranami délky  $d$ . Průsečík strany  $BC$  trojúhelníku  $SBC$  a úsečky  $SO$  označíme jako  $P$ . Tento průsečík  $P$  je středem strany  $BC$  a zároveň patou výšky z vrcholu  $S$ . Velikost  $SP$  bude tedy podle Pythagorovy věty  $\sqrt{d^2 - (1/2 * d)^2} = d\sqrt{3/4}$ .
3. Zároveň platí, že dvě strany rovnostranného trojúhelníka svírají úhel  $60^\circ$ .
4. Použijeme-li pro zadání natočení obrazce v rovině jednotkový vektor  $\mathbf{v}$ , pak platí, že úsečka  $SO$  je lineárním násobkem vektoru  $\mathbf{v}$ . Body  $A, B, C, D$  leží na úsečkách kolmých na vektor  $\mathbf{v}$ , tedy i kolmých na úsečku  $SO$ .

## 4.2 Zadání vektoru větru

K natočení obrazce v rovině souřadnicového systému se používá jednotkový vektor, reprezentující směr proudění vzduchu. Původně jsem zamýšlel použít kompasové zadání směru větru, vznik vektoru pro tento případ ukazuje následující obrázek 4.2:



Obr. 4.2: Kompasové určení směru větru

Uživatel zadá do programu písmennou zkratku (např. JV) reprezentující směr větru, podle obr. 4.2 vane jihozápadní vítr, tzn. vzduch proudí od jihovýchodu k severozápadu, z čehož vyplývá, že se nebezpečná látka bude šířit severozápadním směrem. Program této zkratce přiřadí jeden z předem stanovených 16 úhlů, zde zvýrazněný úhel  $135^\circ$ , a následně vypočítá přes funkce sinus a cosinus souřadnice jednotkového vektoru.

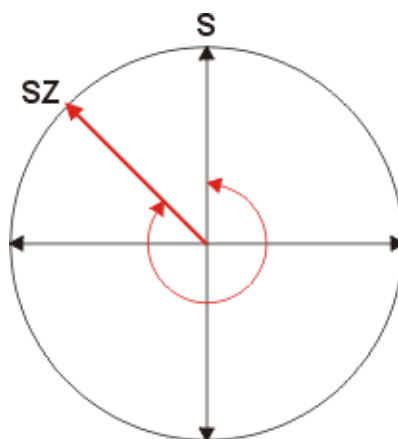
Toto řešení se ovšem neukázalo jako vhodné, jednak pracovalo pouze s omezenými hodnotami úhlů a jednak neuvažovalo stočení os souřadnic referenčního systému oproti

zeměpisným. Při malé velikosti obrazce nebylo toto stočení znatelné, ale při větších rozměrech zasahoval nepootočený obrazec jiné území.

Z výše uvedených důvodů jsem zvolil řešení, kdy se zadává azimut směru proudění vzduchu, tzn. směr kam vítr vane. K tomuto azimutu je nutno pro systém S-JTSK interně přidávat hodnotu meridiánové konvergence, abych získal skutečný směr v souřadnicovém systému, jelikož tato hodnota meridiánové konvergence dosahuje v západní části ČR až 10°. Konvergence se dopočítává v případě, že uživatel při spuštění programu zvolí výsledný souřadnicový systém S-JTSK nebo pokud pro souřadnice místa havárie platí  $X < Y$ ,  $Y < 0$ ,  $X < 0$ . Tyto nerovnosti platí pro souřadnice na území ČR v systému S-JTSK pro GIS (SJTSK\_East\_North), porovnání souřadnic jsem ponechal z důvodu možnosti výběru systému „Jiný“, který se volí tehdy, nezná-li uživatel souřadnicový systém vstupní vrstvy, ale přitom je vrstva v S-JTSK. Pro výpočet konvergence používám vzorec ze [7], kde  $Y$ ,  $X$  jsou souřadnice v km a výsledné  $C$  je ve stupních:

$$C = 0,008257 \cdot Y + 2,373 \cdot \frac{Y}{X}$$

Způsob zadání azimutem ukazuje následující obrázek 4.3, kdy je červeně označen zadávaný azimut a následně vypočtený vektor směru, v tomto případě je azimut roven 315°, lze ale zadat i azimut - 45°.



Obr. 4.3: Zadání azimutem

### 4.3 Zadání místa události

Zadání souřadnic místa havárie je umožněno dvojím způsobem, jednak ruční zadání souřadnic, jednak pomocí jednobodové vektorové vrstvy. V druhém případě uživatel vytvoří nad podkladovou mapou novou vrstvu s jedním bodem vyjadřující místo nehody. Algoritmus si z této vrstvy převezme souřadnice bodu a použije je do výpočtu. Informace o souřadnicích místa nehody nesou i výstupní polygony v atributové tabulce.

### 4.4 Zadání velikosti obrazce

Parametry pro konečnou konstrukci obrazce na obr. 4.1 označené jako  $z$  a  $d$  jsou tabelované hodnoty vzdáleností uvedené v tabulce Table1 příručky ERG2008 (kap 2.1.1). Parametru  $z$  odpovídá *First Isolate*, parametru  $d$  odpovídá *Downwind Distance*. Ručním

zadáním těchto parametrů je ovšem možné upravit velikost obrazce a přizpůsobit ho tím požadované situaci, viz výše uvedené např. rychlý únik velkého množství, či vysoká zástavba. Jelikož je ovšem většinou potřeba znát tabelované hodnoty, uchovávám je v textovém souboru, ze kterého se při spuštění podle druhu látky a dalších parametrů vyhledají (viz dále) a použijí pro výpočet.

## 4.5 Postup výpočtu významných bodů obrazce

Vlastností obrazce uvedených v odstavci 4.1 jsem využil při výpočtu bodů  $O$ ,  $A$ ,  $B$ ,  $C$ ,  $D$ . Na začátku je znám bod  $S$ , jednotkový vektor  $\mathbf{v}$ , hodnota  $d$  a  $z$ . Bod  $O$  získám posunutím bodu  $S$  o  $d$ -násobek vektoru  $\mathbf{v}$ . Stejně tak i bod  $P$  posunutím o násobek  $d' = d \cdot \sqrt{3/4}$ . Vytvořím-li k vektoru  $\mathbf{v} = (v_x, v_y)$  kolmý vektor  $\mathbf{v}_{k1} = (-v_y, v_x)$  a  $\mathbf{v}_{k2} = (v_y, -v_x)$ , získám bod  $A$  posunutím bodu  $S$  o  $z$ -násobek vektoru  $\mathbf{v}_{k1}$ , bod  $D$  posunem  $S$  o  $z$ -násobek vektoru  $\mathbf{v}_{k2}$ . Pomocí stejných vektorů  $\mathbf{v}_{k1}$ ,  $\mathbf{v}_{k2}$ , ovšem z bodu  $P$  a násobku  $d$ , získám body  $B$  a  $C$ . Výše uvedené posuny vyjadřují následující rovnice (1) pro souřadnici  $x$  a (2) pro souřadnici  $y$ :

$$\begin{aligned} P_x &= S_x + \sqrt{0,75} \cdot d \cdot v_x & P_y &= S_y + \sqrt{0,75} \cdot d \cdot v_y \\ O_x &= S_x + d \cdot v_x & O_y &= S_y + d \cdot v_y \\ A_x &= S_x + z \cdot (-1) \cdot v_y & A_y &= S_y + z \cdot (+1) \cdot v_x \\ B_x &= P_x + 0,5 \cdot d \cdot (-1) \cdot v_y & B_y &= P_y + 0,5 \cdot d \cdot (+1) \cdot v_x \\ C_x &= P_x + 0,5 \cdot d \cdot (+1) \cdot v_y & C_y &= P_y + 0,5 \cdot d \cdot (-1) \cdot v_x \\ D_x &= S_x + z \cdot (+1) \cdot v_y & D_y &= S_y + z \cdot (-1) \cdot v_x \end{aligned} \quad (1) \quad (2)$$

Mezilehlé body na kružnici a výseči získám pomocí polárních souřadnic. Vypočítám si směrnik z bodu  $S$  na bod  $D$  a postupným odečítáním  $180^\circ$  od směrniku po daném kroku  $3^\circ$  získám body na půlkružnici  $DA$ . Stejně tak i body na půlkružnici  $AD$  odečítáním  $180^\circ$  od směrniku na bod  $A$ . Tím získám seznam souřadnic bodů kružnice ve směru hodinových ručiček. Body oblouku výseče získám stejným způsobem, odečítáním  $60^\circ$  po daném kroku  $1^\circ$  od směrniku z bodu  $S$  na bod  $B$ . Výše uvedené postupy vyjadřují následující rovnice (3) pro půlkruh  $DA$ , (4) pro půlkruh  $AD$  a (5) pro oblouk výseče  $BC$ :

$$\begin{aligned} \sigma_{SD} &= \arctg \frac{D_y - S_y}{D_x - S_x} & \sigma_{SA} &= \arctg \frac{A_y - S_y}{A_x - S_x} \\ \forall i \in \langle 0; -60 \rangle & & \forall i \in \langle 0; -60 \rangle & \\ \alpha &= i \cdot \frac{180^\circ}{60} & \alpha &= i \cdot \frac{180^\circ}{60} \\ P_{xi} &= S_x + z \cdot \cos(\sigma_{SD} + \alpha) & Q_{xi} &= S_x + z \cdot \cos(\sigma_{SA} + \alpha) \\ P_{yi} &= S_y + z \cdot \sin(\sigma_{SD} + \alpha) & Q_{yi} &= S_y + z \cdot \sin(\sigma_{SA} + \alpha) \end{aligned} \quad (3) \quad (4)$$

$$\sigma_{SB} = \arctg \frac{B_y - S_y}{B_x - S_x}$$

$$\forall i \in \langle 0; -60 \rangle$$

$$\alpha = i \quad (5)$$

$$R_{xi} = S_x + d \cdot \cos(\sigma_{SB} + \alpha)$$

$$R_{yi} = S_y + d \cdot \sin(\sigma_{SB} + \alpha)$$

Obrazec vzniká uložením posloupnosti souřadnic bodů ve směru hodinových ručiček do geometrické struktury *LinearRing* a následně *Polygon* zprostředkované knihovnou JTS od společnosti Vivid Solutions (viz [10]), která je nutnou součástí knihovny Sextante (viz kap. 3.5). Polygon kruhu vzniká uložením seznamu souřadnic bodů půlkruhů *DA* a *AD*. Polygon výseče vzniká uložením přímky *AB*, oblouku *BC*, přímky *CD* a půlkruhu *AD*. První tři části jsou v původním vypočítaném pořadí, půlkruh *AD* je v obráceném pořadí, aby byl zachován nutný jednotný směr posloupnosti bodů ve směru hodinových ručiček.

## 4.6 Uložení a načítání tabelovaných hodnot z příručky ERG 2008

V současnosti obsahuje tabulka Table 1 (viz kap. 2.1.1, obr. 2.3) příručky ERG2008 cca 600 látek. Tabulka je dostupná ve formátu xls (Microsoft Excel) ze stránek příručky [1], z důvodu snadné editace a případného doplnění hodnot, byl jako jeden z požadavků HZS uložit tabelované hodnoty do textového souboru. Data proto uchovávám v textovém souboru, kde jsou jednotlivé hodnoty pro jednu látku odděleny dělicím znakem (středník), který usnadňuje jejich zpracování při vyhledávání. Látky jsou uloženy po řádcích stejně jako v původní tabulce v ERG. Uživatel má možnost zadat jeden nebo druhý vyhledávací klíč, v tomto případě ID (UN kód) nebo název látky (přesný název jako v ERG2008). Hodnoty vzdáleností jsou uloženy pouze v metrech, oproti příručce, kde jsou v metrech, stopách, kilometrech a mílích.

Struktura řádku textového souboru je následujícího vzoru:

*ID; Název; IzolMalý; DistDenMalý; DistNocMalý; IzolVelký; DistDenVelký; DistNocVelký;*

kde *Izol* (= First Isolate) a *Dist* (= Downwind Distance) označuje velikost zóny, *Den* a *Noc* denní dobu a *Malý* a *Velký* rozsah havárie. Uvádím příklad jednoho řádku:

1017;Chlorine;60;400;1600;600;3500;8000;

Jelikož jsou hodnoty po načtení uloženy k atributům obrazce, není nutné uchovávat celou tabulku dále v paměti. Načítání je proto vyřešeno tak, že algoritmus čte po řádcích soubor, každou řádku rozdělí pomocí dělicího znaku a porovnává daný hledaný klíč s načteným ID a názvem, pokud nalezne shodu alespoň v jednom případě, vyhledá ve zbytku řádku podle rozhodovacích podmínek správné rozměry. Čtení je ukončeno, je-li nalezena shoda, nebo pokud je přečten celý soubor. Přestože tento způsob čtení a vyhledávání má lineární složitost, není z důvodu množství dat nutné vkládat látky do jiné struktury s nižší časovou složitostí.

## 4.7 Souřadnicový systém

Jelikož Sextante neumí implicitně pracovat se souřadnicovými systémy a uživatelem vytvořená vstupní vrstva nenese informaci o svém souřadnicovém systému, je nutné systém přiřadit jinak. Aby bylo možné výstupní vrstvu správně zobrazit i v jiných GIS programech. V mém případě jsem tento problém vyřešil přes dotaz na uživatele. Jedním ze vstupních parametrů zadávaným uživatelem je právě výběr z nabídky souřadnicových systémů, ve kterém je vstupní vrstva a bude tedy i výstupní. Nabídka obsahuje 4 možnosti výběru: UTM 33. zóna, UTM 34. zóna, S-JTSK, jiný. Při zvolení jedné z prvních tří možností se k výstupnímu souboru *shp* vygeneruje i soubor *prj* nesoucí informaci o referenčním souřadnicovém systému. Při zvolení čtvrté možnosti se daný soubor nevygeneruje. Protože zadavatel předpokládá další používání výstupní vrstvy převážně v systému ArcGIS, předpokládám tedy export vrstvy do podoby shapefile. Použil jsem proto pro podobu souboru *prj* právě definice společnosti ESRI ze stránek [8]. Tento způsob referencování má bohužel i svá úskalí, jednak má omezenou škálu systémů (3), která by ovšem byla rozšiřitelná, ale hlavně předpokládá poučeného uživatele, který zná souřadnicový systém vstupní vrstvy, případně podkladové mapy, ze které vstupní vrstvu vytváří. Důvodem pro zvolení právě těchto souřadnicových systémů bylo, že zadavatel používá podkladové mapy vojenské i civilní, které mají především tyto souřadnicové systémy. Dalším důvodem bylo, že mapové podklady v těchto systémech lze získat i přes WMS služby.

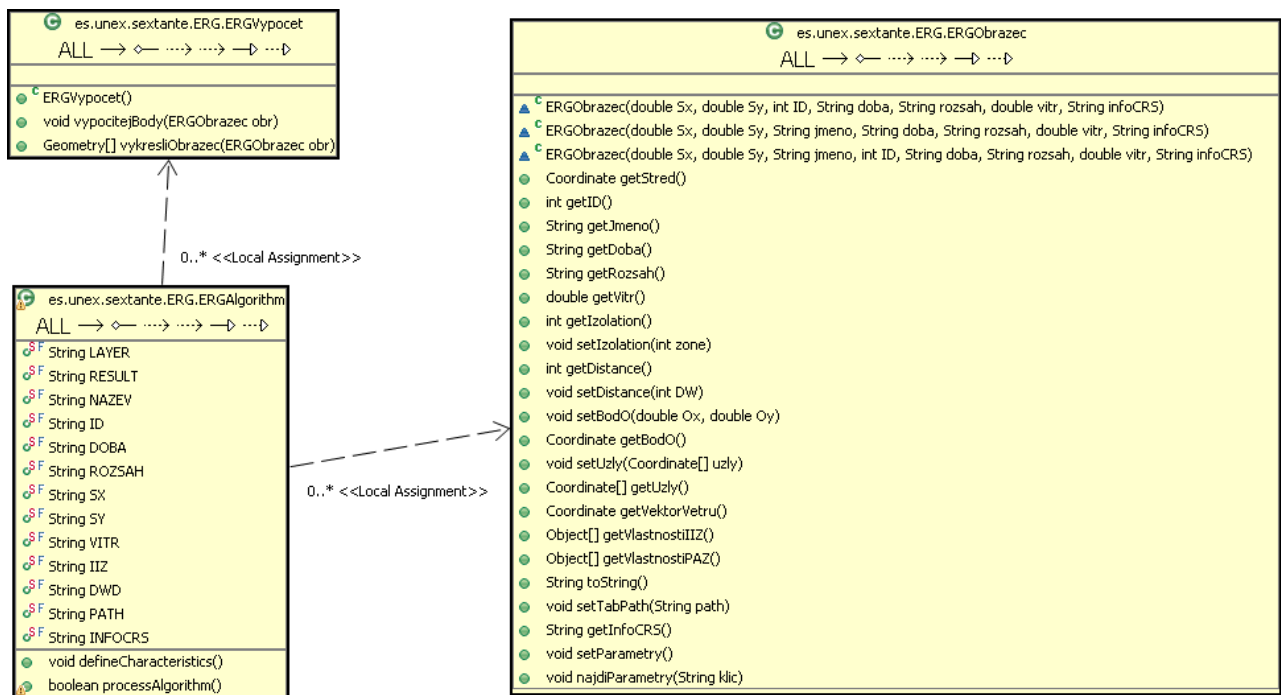
## 4.8 Struktura vytvořeného programu

Program se skládá ze tří tříd. Hlavní třída algoritmu, pojmenovaná podle konvence Sextante jako *ERGAlgorithm*, zařazuje algoritmus do struktury Toolbox, nastavuje podobu vstupního okna pro zadávání uživatelských parametrů, zakládá třídu *ERGObrazec* podle hodnot vstupních parametrů, volá třídu *ERGVypocet* pro výpočet bodů a podle volby uživatele vytváří soubor *prj* s definicí souřadnicového systému. Po výpočtu obrazce ukládá polygony do výstupní vrstvy s jejich atributy a ukončuje běh programu.

Třída *ERGObrazec* slouží pro uchování všech parametrů obrazce, interně přidává případnou meridiánovou konvergenci k azimutu větru a vypočítává jednotkový vektor směru větru. Vlastní metodou také vyhledává v souboru s tabelovanými hodnotami. Sestavuje také řetězce pro výstupní atributovou tabulku.

Třída *ERGVypocet* slouží k samotným výpočtům bodů obrazce a následnému uložení seznamu souřadnic do jednotlivých geometrických struktur vytvářející polygony.

Strukturu programu ukazuje také následující UML (Unified Modeling Language) diagram (obr. 4.4), generovaný programem Green UML [9].



Obr. 4.4: UML diagram programu

## 4.10 Vstupní okno uživatelského rozhraní

Po spuštění algoritmu přes Toolbox se zobrazí okno s nabídkou vstupních parametrů (viz obr. 4.5). Položky okna budou popisovat odshora:

1. **Místo nehody** – rozbalovací nabídka se seznamem bodových vrstev načtených v GIS programu, při zvolení položky [not selected] je nutné zadat souřadnice místa nehody ručně
2. **Souradnicovy system vstupu i vystupu** – rozbalovací nabídka se 4 možnostmi souřadnicových systémů: UTM 33 zóna, UTM 34 zóna, S-JTSK, jiný.
3. **Nazev latky** – prázdná kolonka pro zadání názvu látky, slouží k vyhledání v tabulce ERG
4. **ID latky** – kolonka pro zadání číselného UN kódu látky, slouží k vyhledání v tabulce ERG
5. **Denni doba** – rozbalovací nabídka pro vybrání denní doby události, obsahuje 2 možnosti: Den, Noc
6. **Rozsah nehody** – rozbalovací nabídka pro vybrání rozsahu nehody, obsahuje 2 možnosti: Malý, Velký
7. **Azimut vetru** – kolonka pro číselné zadání azimutu vanoucího větru v rozsahu +360° až -360°
8. **Souradnice X** – kolonka pro ruční zadání souřadnice místa nehody
9. **Souradnice Y** – kolonka pro ruční zadání souřadnice místa nehody



10. **Soubor tabulek** – kolonka buďto pro ruční vypsání cesty k textovému souboru, nebo vyhledání v adresářové struktuře přes kliknutí na ikonu u pravého okraje okna
11. **Izolacni vzdalenost** – kolonka pro ruční vepsání rozměru nebezpečné zóny
12. **Vzdalenost po vetru** – kolonka pro ruční vepsání rozměru ohrožené zóny
13. **Vizualizace zony polygon** – kolonka pro zadání místa uložení a jména výstupního souboru, vyhledání v adresářové struktuře pomocí ikony u pravého okraje okna

Obr. 4.5: Vstupní okno programu

V případě výběru vstupní vrstvy v bodě 1 není nutné zadávat souřadnice místa nehody v bodě 8 a 9. Při zadávání identifikace je možné zadat buď jen číselné ID, nebo jen název, nebo oboje. Při načtení souboru s tabelovanými hodnotami v bodě 10, není nutné ručně zadávat vzdálenosti v bodech 11 a 12.

## 4.11 Výstupní polygony

Po úspěšném dokončení algoritmu se zobrazí výstupní vrstva s názvem „Vizualizace zon“ se dvěma polygony, reprezentující jednotlivé zóny v místě havárie. Prvním polygonem je nebezpečná zóna („Initial Isolation Zone“, obr. 4.6, kap 2.1.1), která má svůj řádek v atributové tabulce s těmito hodnotami:

*Jmeno, Cislo ID, Denní doba, Rozsah, Azimut\_vitr, X-stredu, Y-stredu, Polomer ,*

kde *Jmeno* a *Cislo ID* identifikuje uniklou látku, *Denní doba*, *Rozsah*, *Azimut\_vitr* charakterizují podmínky v místě havárie, *X-stredu*, *Y-stredu* jsou souřadnice místa havárie a *Polomer* udává rozměr zóny.



Obr.4.6: Nebezpečná zóna

Druhým polygonem je zóna ohrožení („Protective Action Zone“, obr. 4.7, kap. 2.1.1), která má v atributové tabulce stejné veličiny jako nebezpečná zóna:

*Jmeno*, *Cislo ID*, *Denní doba*, *Rozsah*, *Azimut\_vitr*, *X-stredu*, *Y-stredu*, *Polomer*,



Obr. 4.7: Ohrožená zóna

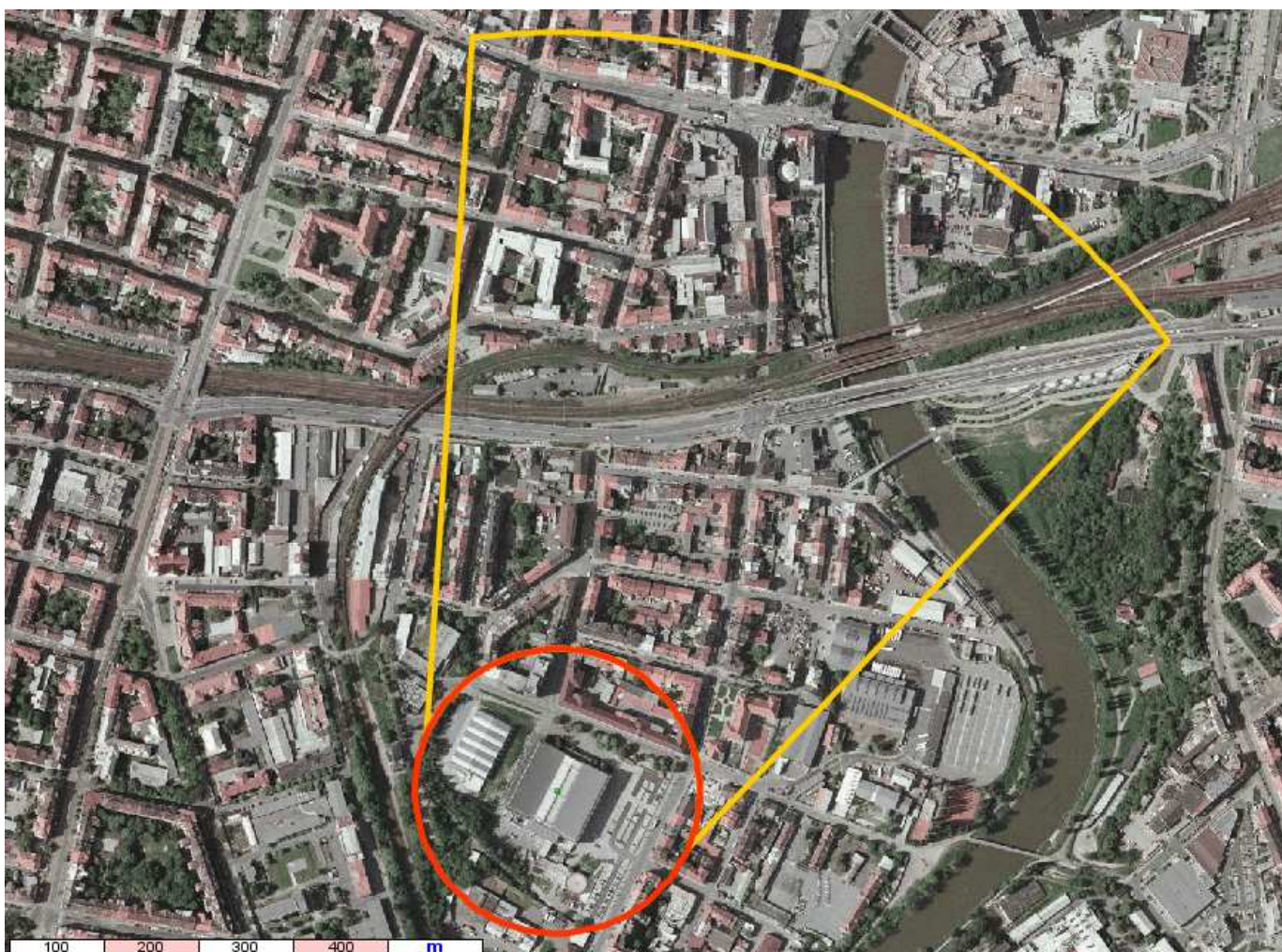
Rozdělení výstupu do dvou polygonů byl jeden z požadavků zadavatele, aby bylo možno zobrazovat zóny i jednotlivě, jelikož na jejich území každé z nich bude probíhat rozdílná činnost zasahujících jednotek.

## 4.12 Shrnutí implementace

Výsledkem zpracování je algoritmus implementovaný do knihovny Sextante, která po přidání do některého z podporovaných GIS programů umožňuje jeho použití. Algoritmus je uložen v jar archivu *sextante\_ERG.jar*, tento archiv se pouze dokopíruje k instalaci Sextante. Potřebnou součástí je i soubor *ERG\_komplet.txt*, obsahující kompletní tabelované hodnoty z příručky ERG 2008 v podobě čitelné algoritmem (vzor v kap. 4.6).

Po spuštění algoritmu je uživatel dotázán na nutné parametry a okolnosti nehody (viz odstavec 4.10 a obr. 4.5), po výpočtu algoritmu se jako výsledek na obrazovce zobrazí nová vektorová vrstva se 2 polygony (viz odstavec 4.11 a obr. 4.6 a 4.7). Vrstvu

doporučuji ukládat primárně jako shapefile, kdy je možno využít referenčního souboru *prj*. Ukázkový výstup představuje následující obrázek 4.8.



Obr. 4.8: Ukázkový výstup nad ortofotem

Na obr. 4.8 je fiktivní únik velkého množství amoniaku ze zimního stadionu v Plzni za dne při větru vanoucímu přibližně severním směrem. Červený kruh je oblast nebezpečné zóny, žlutý znázorňuje zónu ohrožení. Podkladem je ortofotomapa o velikosti pixelu 1 m získaná přes WMS službu z České informační agentury životního prostředí. Podobu atributové tabulky předchozí vizualizace znázorňuje obrázek 4.9.

Attributes: Vizualizace zon								
Vizualizace zon (2 Features)								
FID	Jmeno	Cislo_ID	Denni_doba	Rozsah	Azimut_vitr	X-stredu	Y-stredu	Polomer
15	Ammonia, anhydrous	1005	Den	Velky	23.614874266720623	-822584.0232773123	-1070876.697073749	150
16	Ammonia, anhydrous	1005	Den	Velky	23.614874266720623	-822584.0232773123	-1070876.697073749	800

Obr. 4.9: Atributová tabulka

## 5 Závěr

Výsledkem této práce je funkční program vykreslující rozsah zasaženého území při havárii podle algoritmu z příručky ERG2008. Algoritmus je implementován jako součást open source knihovny geoalgoritmů, která je připojitelná k řadě open source GIS programů (viz kap. 3), jejichž výstupem jsou také formáty souborů použitelné v dalších i komerčních GIS softwarech. Správnou funkčnost algoritmu je samozřejmě nutné ještě řádně testovat a porovnat jeho výsledky s podobnými programy používanými na HZS, než bude možné tento program běžně používat.

Výstupní vrstva bude využitelná při tvorbě tematických map zasaženého území, případně jako vstupní parametr dalších prostorových analýz. Například k analýze počtu osob, který je nutný evakuovat ze zasaženého území nebo k analýze zasažených komunikací, odkud je nutné odklonit dopravu.

Další výsledkem jsou také zkušenosti s prací v prostředí knihovny Sextante. Po těchto zkušenostech mohou knihovnu Sextante doporučit jako prostředí pro implementaci prostorových algoritmů.

## Literatura

- [1] CLOUTIER, M. – CUSHMAC, G.: *Emergency Response Guidebook 2008*. Transport Canada. U.S. Department of Transportation, Secretariat of Communications and Transport of Mexico. 2008. Dostupné z URL <http://www.phmsa.dot.gov/hazmat/library/erg>.
- [2] *Bojový řád jednotek požární ochrany – taktické postupy zásahu*. Ministerstvo vnitra – generální ředitelství Hasičského záchranného sboru České republiky. Praha: 2004. Kapitola L. [online]. [citováno 20. 1. 2009].  
Dostupné z URL [http://web.mvcr.cz/archiv2008/hasici/izs/bojrad/index\\_hasici.html](http://web.mvcr.cz/archiv2008/hasici/izs/bojrad/index_hasici.html)
- [3] *Bojový řád jednotek požární ochrany – taktické postupy zásahu*. Ministerstvo vnitra – generální ředitelství Hasičského záchranného sboru České republiky. Praha: 2002. Kapitola Ř. Metodický list 7. [online]. [citováno 20. 1. 2009]  
Dostupné z URL [http://web.mvcr.cz/archiv2008/hasici/izs/bojrad/index\\_hasici.html](http://web.mvcr.cz/archiv2008/hasici/izs/bojrad/index_hasici.html)
- [4] *Sextante*. Poslední úprava 25. 2. 2009. [citováno 20. 4. 2009].  
URL adresa <http://www.sextantegis.com>
- [5] OLAYA, V.: *Introduction to SEXTANTE*. 2008. Edition 1.0. Rev. 10. 9. 2008.  
Dostupné z URL <http://www.sextantegis.com>.
- [6] OLAYA, V.: *SEXTANTE Programming Guide*. 2008. Edition 1.0. Rev. 4. 11. 2008.  
Dostupné z URL <http://www.sextantegis.com>.
- [7] BARANOVÁ, M.: *Multimediální texty Matematické kartografie*. 2004. aktualizace 2. 3. 2005. [citováno 4. 5. 2009]. (online). Dostupné z URL:  
[http://www.gis.zcu.cz/studium/mk2/multimedialni\\_texty/index\\_soubory/index.html](http://www.gis.zcu.cz/studium/mk2/multimedialni_texty/index_soubory/index.html)
- [8] *Spatial Reference*. [citováno 15. 5. 2009]. URL adresa <http://spatialreference.org/>.
- [9] *Green UML*. [software]. URL adresa <http://green.sourceforge.net/>
- [10] *Java Topology Suite*. [citováno 20. 2. 2009].  
URL adresa <http://www.vividsolutions.com/jts/jtshome.htm>
- [11] AQUINO, J.: *JTS Topology Suite – Developer's Guide*. Verze. 1.4. Datum vydání 17. 10. 2003. Dostupné z URL <http://www.vividsolutions.com/jts/jtshome.htm>.
- [12] *Wikipedie, otevřená encyklopedie*. URL adresa <http://cs.wikipedia.org>

## Obsah příloženého CD

