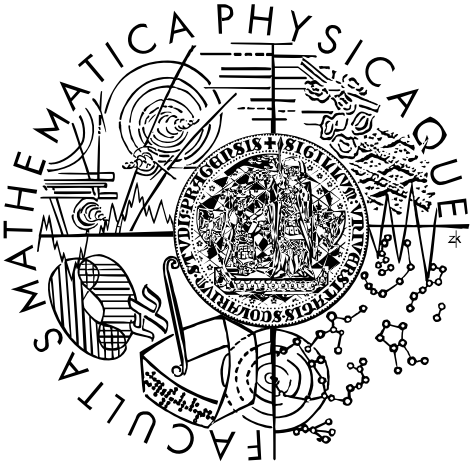
Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Šimon Rajčan

Adaptivní klasifikátor pošty pro IMAP servery

Katedra softwarového inženýrství

Vedoucí bakalářské práce: RNDr. David Bednárek

Studijní program: Informatika, Obecná informatika

2009

Na tomto mieste by som sa chcel poďakovať svojej rodine, za to že ma počas štúdia podporovala a svojmu vedúcemu RNDr. Davidovi Bednárekovi za odborné vedenie.

Prehlasujem že som svoju bakalársku prácu napísal samostatne a výhradne s použitím citovaných prameňov. Súhlasím s zapožičaním práce a jej zverejňovaním.

V Prahe dňa 6.8.2009 Šimon Rajčan

Obsah

Název práce: Adaptivní klasifikátor elektronické pošty pro IMAP servery

Autor: Šimon Rajčan
Katedra: Katedra softwarového inženýrství
Vedoucí bakalářské práce: RNDr. David Bednárek
e-mail vedoucího: david.bednarek@mff.cuni.cz

Abstrakt: Předmětem této práce je navrhnout a implementovat systém pro klasifikaci došlé elektronické pošty na základě pravidel získaných během práce uživatele v učícím režimu aplikace. Základním cílem je pozitivní vyhledávání užitečné/významné pošty, nikoliv odstranění nežádoucí pošty. Systém bude implementován jako interaktivní aplikace pracující pomocí protokolu IMAP4 s poštou uloženou na serveru.

Klíčová slová: IMAP, Bayes, filter, elektronická pošta

Title: Adaptive classifier of electronic mail for IMAP servers

Author: Šimon Rajčan
Department: Department of Software Engineering
Supervisor: RNDr. David Bednárek
Supervisor's e-mail address: david.bednarek@mff.cuni.cz

Abstract: The subject of this work is to project and implement a system for classification of certified electronic mail, based on rule acquired in certain mode of application. The main aim is classification of positive mail, not abstraction of ineligible mail. System will be implemented as interactive application worked with protokol IMAP4 with mail stored on server.

Keywords: IMAP, Bayes, filter, electronic mail

Kapitola 1

Úvod

S rozrastajúcim sa Internetom sa elektronická pošta stáva stále častejšie využívaným prostriedkom na komunikáciu. Každý deň si touto lacnou formou vymieňajú správy milióny ľudí. Poštové schránky sú preto často preplnené množstvom nežiadúcich, či nedôležitých správ (SPAM-ov). Preto vzniklo mnoho programov (tzv. SPAM-filtrov), ktoré dokážu SPAM rozpoznať a odstrániť. Čomu sa však prikladá menšia dôležitosť je fakt, že aj po odstránení nežiadúcej pošty ostávajú mnohé schránky plné takej pošty, ktorá nie je SPAM-om, ale tiež nie je pre používateľa dôležitá. V takom prípade je pre používateľa zdĺhavé nájsť vo veľkom množstve pošty takú, ktorá je pre neho skutočne dôležitá, prípadne na ňu treba odpovedať ihneď. A práve týmto problémom sa zaoberá táto práca.

Predmetom tejto práce je navrhnúť a implementovať adaptívny klasifikátor došlej elektronickej pošty, ktorý sa nebude primárne zameriavať na rozpoznávanie nežiadúcich, ale naopak, na rozpoznávanie dôležitých správ. Bude to desktopová aplikácia pracujúca pod systémom Windows. Bude mať tri módy. Prvý je mód učenia, v ktorom bude používateľ ručne označovať, ktorá správa patrí do ktorej kategórie. Druhý mód je mód učenia, v ktorom si aplikácia sama zistí, že ktorá správa patrí do ktorej kategórie na základe predchádzajúcich akcií užívateľa v jeho poštovom klientovi. Tretí mód je mód triedenia, v ktorom bude program pomocou informácií získaných z prvého a druhého módu a použitím vhodného triediaceho algoritmu schopný sám rozdeliť došlú poštu.

Aplikácia bude sťahovať poštu priamo z poštového serveru. Najpoužívanejší internetový protokol na sťahovanie pošty zo serveru je Post Office Protocol version 3 (POP3). POP3 nevyžaduje trvalé pripojenie a je pomerne jednoduchý na ovládanie, avšak nevie pracovať s adresármi. Preto je nutné použiť protokol Internet Message Access Protocol (IMAP), ktorý umožňuje plnú prácu s poštovou schránkou.

Jeho hlavné nevýhody sú, že je zložitý na ovládanie a nepodporuje ho väčšina poštových serverov. V súčasnosti sa používa verzia IMAP4.

Kapitola 2 sa zaoberá voľbou triediaceho algoritmu a jeho podrobným popisom. Kapitola 3 popisuje architektúru programu a zdôvodňuje výber knižnice na prácu s protokolom IMAP.

V Kapitolách 4 a 5 je užívatelská a programátorská dokumentácia.

Kapitola 6 je záver, nasleduje zoznam použitej literatúry a neoddeliteľnou súčasťou práce je aj priložené CD s programom.

Kapitola 2

Metódy triedenia

2.1 Voľba druhu filtra

Voľba vhodného algoritmu je zrejme najdôležitejšia časť celej práce, a preto je tomuto problému nutné venovať najviac pozornosti. V nasledujúcich odstavcoch až po kapitolu 2.2 sú informácie čerpané z [1].

Pri detekcii SPAMu sa najčastejšie používajú dva typy filtrov.

Prvým typom sú filtre založené na určitých pravidlách. Tieto filtre vyhľadávajú v správach rysy, ktoré sú pre spam typické. Ide jednak o niektoré slová (napr. Viagra), príp. slovné spojenia, a o chyby, ktoré sú pre spam typické. Príkladom takýchto chýb je napríklad dátum odoslania v budúcnosti, nedovolené znaky v hlavičke, chybne označený MIME-typ správy a podobne. Za každý rozpoznaný rys je správe pridelené bodové ohodnotenie, body sa potom sčítajú a podľa toho, či výsledný súčet presiahne istú hranicu, je správa pokladaná za spam.

Druhým typom sú filtre, ktoré sú založené na učení (často označované jako Bayesovské). V režime učenia sa filtru predkladajú správy explicitne označené ako spam, či ham (opak spamu), filter z nich vytiahne informácie, ktoré si uloží. Tieto informácie sú najčastejšie slová (príp. iné časti textu), pre ktoré štatisticky zisťuje pravdepodobnosť, že správa, ktorá toto slovo obsahuje, je spam. V režime rozpoznávania potom filter využíva nazhromaždené informácie a testovanej správe priradí pravdepodobnosť, či sa jedná alebo nejedná o spam. Väčšinou sa pre výpočet pravdepodobnosti používa vzorec, ktorý navrhol anglický matematik Bayes.

Prvý typ je pre túto prácu nevhodný, pretože pri poslaní legitímnej správy sa v nej zriedkakedy vyskytujú chyby, aké sú popísané vyššie. Preto bol zvolený druhý typ. Bayesovské filtre určené na detekciu spamu majú hlavnú nevýhodu v tom,

že producenti spamu (spameri) do svojich správ často vkladajú slová, o ktorých vedia, že sa často vyskytujú v legitímnych správach s úmyslom filter oklamať. Táto práca sa však zaoberá klasifikáciou pozitívnej pošty, takže tento problém odpadá. Používateľ, ktorý posiela legitímnu správu do nej prirodzene nevkladá také slová, aby jeho správa bola zaradená do inej kategórie.

2.2 Matematický základ

Bayesovské filtre využívajú Bayesovu teóriu, a to hneď dva krát. Prvý krát pri počítaní pravdepodobnosti, že daná správa je spam, pokiaľ sa v nej nachádza určité slovo a druhý krát, pri počítaní pravdepodobnosti, že sa jedná o spam, ak poznáme pravdepodobnosti pre všetky slová v správe. V nasledujúcich odstavcoch až po kapitolu 2.3 sú informácie čerpané z [2].

2.2.1 Počítanie pravdepodobnosti, že správa obsahujúca dané slovo je spam

Vzorec na počítanie pravdepodobnosti, že správa obsahujúca dané slovo je spam, odvodený z Bayeseovej teórie v základnej forme, vyzerá takto:

$$\Pr(S|W) = \frac{\Pr(W|S)\cdot\Pr(S)}{\Pr(W|S)\cdot\Pr(S)+\Pr(W|H)\cdot\Pr(H)} \quad (1)$$

kde:

Pr(S|W) je pravdepodobnosť, že správa je spam, ak vieme že obsahuje slovo *W*

Pr(W|S) je pravdepodobnosť, že ak je správa spam, tak obsahuje slovo *W*

Pr(S) je celková pravdepodobnosť, že prichádzajúca správa bude spam

Pr(W|H) je pravdepodobnosť, že ak je správa ham, tak obsahuje slovo *W*

Pr(H) je celková pravdepodobnosť, že prichádzajúca správa je ham

2.2.2 Počítanie pravdepodobnosti, že správa je spam

Bayesové filtre predpokladajú, že všetky slová, ktoré sa v správe nachádzajú, sú nezávislé javy. Tento predpoklad nie je vždy správny, pretože napríklad v slovenčine je veľká pravdepodobnosť, že po slovese *„je“* bude nasledovať prídavné meno. Napriek tomu sa v mnohých prípadoch[7,8] ukázalo, že takýto Bayesov filter, je na detekciu spamu veľmi silný nástroj. Samotný vsťah vyzerá nasledovne:

$$p = \frac{p1 \cdot p2 \cdot \ldots \cdot pN}{p1 \cdot p2 \cdot \ldots pN + (1-p1) \cdot (1-p2) \cdot \ldots \cdot (1-pN)} \quad (2)$$

kde:

p je pravdepodobnosť že uvažovaná správa je spam

p1, p2, pN sú pravdepodobnosti pre jednotlivé slová v správe

Takýmto spôsobom sa dá pre správu zistiť, že aká je pravdepodobnosť že patrí do nejakej kategórie (spam, legitímna správa). V našom prípade však nepotrebujeme poznať pravdepodobnosti pre jednotlivé kategórie, ale stačí nám vedieť že pre ktorú kategóriu je táto pravdepodobnosť najvyššia. Taktiež nestačí poštu triediť len na 2 kategórie. Preto bolo od pôvodného vsťahu upustené a za základ algoritmu sa vzal iný Bayesov klasifikátor.

2.3 Iná interpretácia Bayesovho klasifikátora

Ďalšia interpretácia Bayesovho filtra vyzerá nasledovne[3]:

$$V_{Class} = P(Class) \cdot \prod_{i=1}^{N} P(W_i|Class) \quad (3)$$

pričom $P(W_i|Class)$ sa počíta podľa:

$$\frac{1+no(W_i|Class)}{\sum_{y=1}^{N} no(W_y|Class)+|V|} \quad (4)$$

kde:

V_{Class} je hodnota pre triedu *Class.*

P(Class) je pravdepodobnosť že nová správa bude patriť do kategórie *Class*.

$no(W_i|Class)$ je počet výskytov slova *Wi* v správach, ktoré boli pri učení zaradené do triedy *Class*.

$\sum_{y=1}^{N}(W_y|Class)$ je počet výskytov všetkých slov v správach, ktoré boli při učení zaradené do triedy Class.

|*V*| je počet slov, ktoré sa vyskytli v správach pri učení.

V čitateli sa k $no(W_i|Class)$ pripočítava 1, pretože pre slovo v testovacej správe, ktoré pri učení nikdy nebolo zaradené do triedy *Class,* by $no(W_i|Class)$ bola 0. Z toho plynie, že aj V_{Class} by bola 0 a to je nesprávne.

Filter správu zaradí do tej kategórie, pre ktorú bude V_{Class} najväčšie.

2.4 Prispôsobenie Bayesovho spamového filtra na klasifikáciu legitímnych správ

V našom prípade potrebujeme poštu rozdeliť do 2 až X kategórií. Taktiež budeme predpokladať, že pre každú novú správu je rovnaká pravdepodobnosť, že bude patriť do ktorejkoľvek z uvažovaných kategórií.

Výsledný vsťah vyzerá nasledovne:

$$V_{Class=\prod_{i=1}^{N}\frac{1+no(W_i|Class)}{\sum_{y=1}^{N} no(W_y|Class)+|V|}} \quad (5)$$

2.5 Spôsob použitia zvoleného algoritmu na klasifikáciu správ

Vyššie popísaný algoritmus bude použitý na tieto zložky správy:

Telové zložky:

- Telo správy

Hlavičkové zložky:

- Predmet správy
- Adresu odosielateľa
- Doménu odosielateľa
- Mailer odosielateľa (program, ktorý bol použitý na odoslanie správy)

To, do akej kategórie bude nakoniec správa presunutá, sa spočíta takto:

1. Ak filter nedokáže zaradiť správu podľa jej tela (napríklad sa bude jednať o správu s prázdnym telom), bude sa postupovať nasledovne:
    a. Ak filter zaradí správu podľa akýchkoľvek dvoch hlavičkových zložiek do rovnakej kategórie, správa bude presunutá do tejto kategórie. Ak nastane prípad, že filter zaradí podľa dvoch hlavičkových zložiek do dvoch kategórií, tak prednosť má predmet a potom doména odosielateľa.
    b. Ak filter zaradí správu aspoň podľa jednej zložky, správa bude presunutá do tejto kategórie. Prioritu ma predmet, potom doména, potom adresa a nakoniec mailer.
    c. V ostatných prípadoch bude správa presunutá do prvej kategórie.

2. Ak filter dokáže zaradiť správu podľa jej tela, bude sa postupovať nasledovne:
    a. Ak filter zaradí správu aspoň podľa jednej z hlavičkových zložiek do rovnakej kategórie ako telo, správa bude presunutá do tejto kategórie.
    b. Ak filter zaradí správu aspoň podľa troch hlavičkových zložiek do rovnakej kategórie, správa bude presunutá do tejto kategórie. Nesmie sa však jednať o zložky adresa, doména a mailer.
    c. Inak bude správa presunutá podľa zaradenia tela.

Kapitola 3

Architektúra

3.1 Programovací jazyk

Imapová knižnica, ktorá je použitá, nieje prenositeľná, preto nebolo nutné program implementovať napríklad v Jave a ako programovací jazyk zvolený C# na platforme .NET, ktorý umožňuje jednoduchú prácu s formulármi v prostredí Windows.

3.2 Vrstvy programu

Program sa delí na 3 vrstvy.

Prvá vrstva je grafická a je reprezentovaná troma formulármi. Hlavný formulár sa volá FormMain a z neho sa dajú spúšťať ostatné formuláre.

Druhá vrstva sú knižnice, ktoré využíva prvá vrstva. Sú to MyDictionary, MyImap, Filter a Users.

Posledná vrstva je databázová, ktorá spravuje informácie o užívateľoch a informácie získané pri učení aplikácie.

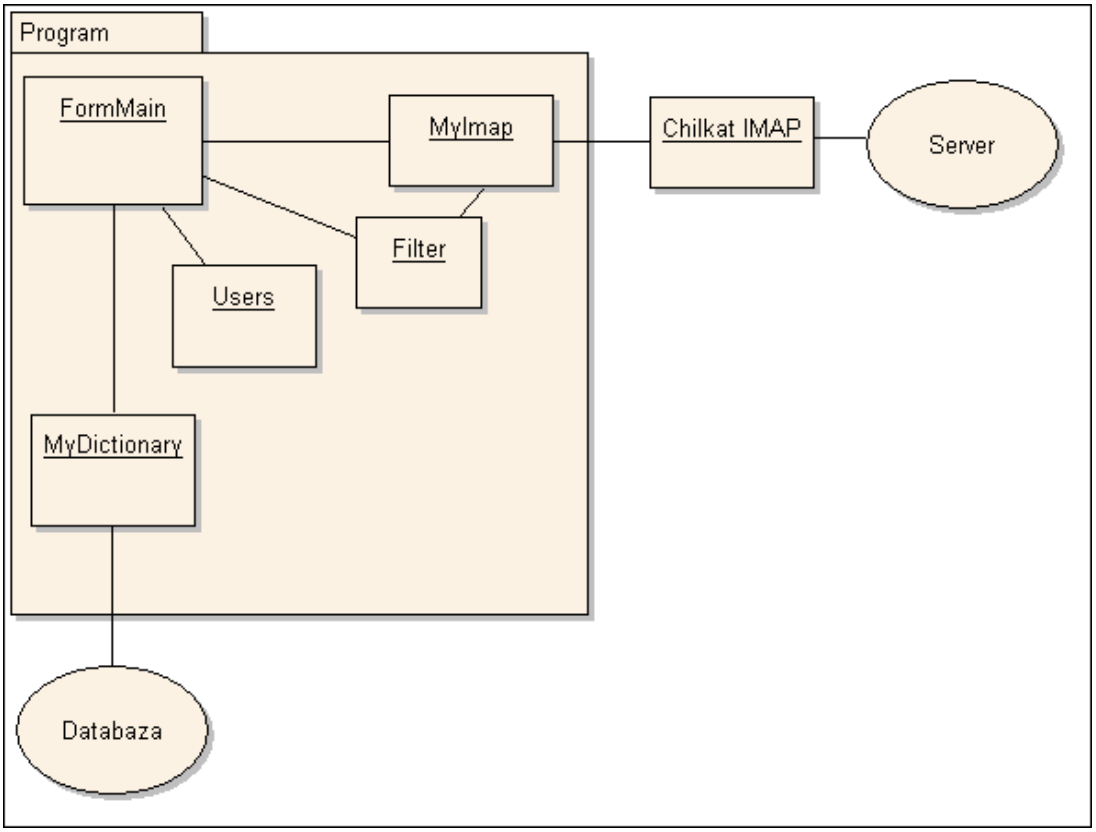
3.3 Členenie programu

Hlavné komponenty programu sú tieto:

- Hlavný formulár (FormMain)
- Komponent na správu dát. Reprezentuje ho trieda MyDictionary.cs
- Komponent na prácu s serverom. Reprezentuje ho trieda MyImap.cs

- Komponent na klasifikáciu správ. Reprezentuje ho trieda Filter.cs
- Komponent na správu používateľov. Reprezentuje ho trieda Users.cs

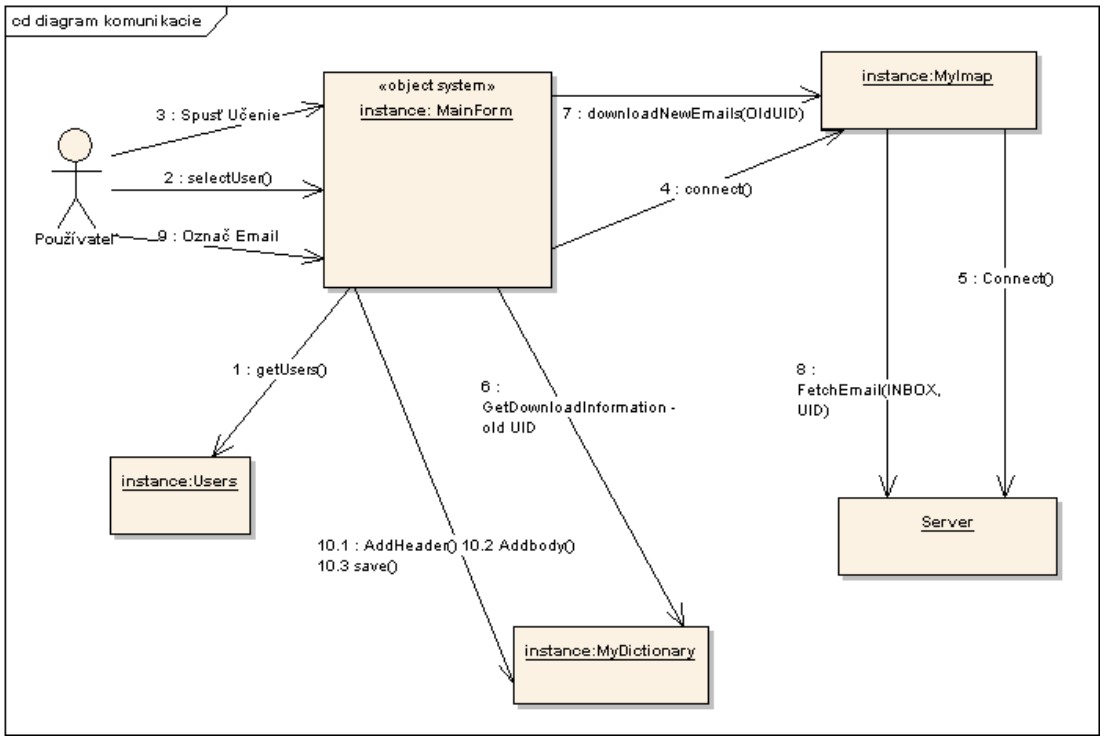
Program okrem databázy a servera komunikuje ešte s knižnicou Chilkat Imap, ktorá sa stará o prácu s protokolom Imap. Členenie programu zobrazuje obrázok 3.1.



Obrázok 3.1 Architektúra

3.4 Mód mechanického učenia

Mód mechanického učenia popisuje obrázok 3.2. Každá šípka predstavuje nejakú udalosť. Udalosti sa vykonávajú v takom poradí, ako sú očíslované.



Obrázok 3.2 Učenie

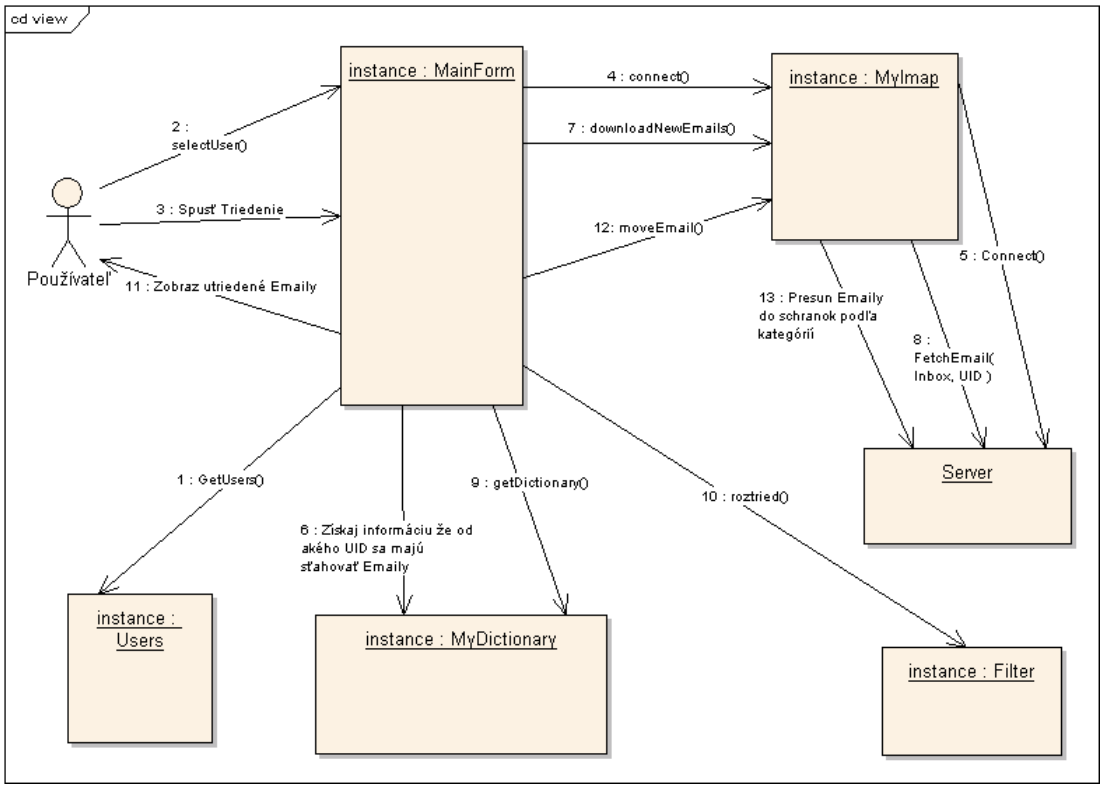
1. FormMain si od Users vypýta zoznam používateľov a zobrazí ho
2. Používateľ vyberie používateľa
3. Používateľ spustí učenie
4. FormMain prikáže MyImap, aby sa pripojil na server
5. MyImap sa pripojí na server
6. MyForm si od MyDictionary vypýta informáciu, o UID poslednej spracovanej správy
7. MyForm prikáže MyImap stiahnuť správy zo schránky Inbox, z vyšším UID ako bolo posledné spracované
8. MyImap stiahne správy zo serveru
9. MyForm zobrazí správu
10. Používateľ označí správu, do ktorej kategórie patrí
11. MyForm prikáže MyDictionary, aby z danej správy uložil potrebné informácie

3.5 Mód automatického učenia

Mód automatického učenia funguje podobne ako mód mechanického učenia.

1. FormMain si od Users vypýta zoznam používateľov a zobrazí ho
2. Používateľ vyberie používateľa
3. Používateľ spustí učenie
4. FormMain prikáže MyImap, aby sa pripojil na server
5. MyImap sa pripojí na server
6. MyForm si od MyDictionary vypýta informáciu, o UID posledných spracovaných správach pre všetky kategórie
7. MyForm prikáže MyImap stiahnuť správy z priečinku, ktorý prináleží prvej kategórii, z vyšším UID ako bolo pre daný priečinok posledné spracované
8. MyImap stiahne správy zo serveru
9. MyForm pre každú správu prikáže MyDictionary, aby z nej uložil potrebné informácie
10. Body 7 až 9 sa opakujú pre každú kategóriu

3.6 Mód triedenia



Obrázok 3.3 Triedenie

Mód triedenia popisuje obrázok 3.3. Každá šípka predstavuje nejakú udalosť. Udalosti sa vykonávajú v takom poradí, ako sú očíslované.

1. FormMain si od Users vypýta zoznam používateľov a zobrazí ho
2. Používateľ vyberie používateľa
3. Používateľ spustí triedenie
4. FormMain prikáže MyImap, aby sa pripojil na server
5. MyImap sa pripojí na server
6. FormMain si od MyDictionary vypýta informáciu, o UID poslednej spracovanej správy
7. FormMain prikáže MyImap stiahnuť správy z priečinku Inbox podľa ich UID
8. MyImap stiahne správy
9. MyForm si od MyDictionary vypýta informácie potrebné na filtrovanie správ

10. MyForm prikáže Filter roztriediť dané správy
11. MyForm zobrazí roztriedené správy
12. MyForm prikáže MyImap presunúť správy na serveri podľa príslušných kategórií
13. MyImap presunie správy na Serveri do príslušných priečinkov

3.7 Komunikácia s Imapovým serverom

3.7.1 Výber knižnice pre prácu s Imapovým serverom

Pri hľadaní knižnice na prácu s Imapom, boli nájdené tieto knižnice

- Chilkat.Imap: Rozsiahla knižnica, ktorá umožňuje plnohodnotnú prácu s Ímapom a stiahnutými správami. Je k nej zrozumiteľná dokumentácia a množstvo vzorových príkladov použitia. Má podporu pre protokol SSL. Spoločnosť Chilkat ponúka množstvo produktov, čo je zárukou stability. Bohužiaľ zadarmo je dostupná iba 30-dňová trial verzia. Viac informácií v [4].

- xemail.net: Jednoduchá a bezplatná knižnica, ktorá umožňuje plnohodnotnú prácu s Imapom. Má podporu pre protokol SSL. Bohužiaľ má slabšiu dokumentáciu a funguje len v prostredí Linux. Viac informácií v [5].

- Imap Client library using C#: Jednoduchá a bezplatná knižnica. Funguje pod Windows. Bohužiaľ neumožňuje plnohodnotnú prácu s Imapom (napríklad presúvanie správ medzi priečinkami). Má slabšiu dokumentáciu. Môže byť nestabilná. Autor sám priznáva, že je to jeho prvý projekt v C#. Viac informácií v [6].

Po zhodnotení všetkých kladov a záporov, bola vybraná knižnica Chilkat.Imap. Hlavné dôvody boli, že obsahuje prehľadnú dokumentáciu, a že umožňuje v súvislosti s Imapom všetko, čo je pre túto prácu potrebné.

3.8 Ukladanie dát

3.8.1 Dáta, ktoré je nutné ukladať

Zo správ, ktoré prejdú procesom „učenia“, je potrebné uložiť vybrané údaje. Pre každého používateľa je nutné si pamätať názov jeho poštového servera, jeho login, heslo (heslo je nepovinné), názvy kategórií, na ktoré chce svoju poštu členiť, u každej kategórii názov priečinku na jeho serveri, do ktorého chce poštu danej kategórie presúvať a to, či používa SSL na porte 933. Taktiež je potrebné pre každého používateľa mať dva slovníky. V prvom slovníku budú uložené informácie získané z tiel správ a to pre každé slovo jeho početnosť v každej z kategórií, počet výskytov všetkých slov v každej kategórii a dátum, od ktorého sa budú nabudúce sťahovať nové správy. V druhom slovníku budú uložené informácie získané z hlavičiek správ. Budú to podobne ako pre každé slovo v prvom slovníku, tak pre každé slovo v predmete správy, adresu odosielateľa, doménu odosielateľa a jeho mailer (program, ktorý používa na odosielanie správ), početnosť v každej kategórii.

3.8.2 Spôsob ukladania dát

Dáta sa budú ukladať do xml súborov. Tieto súbory budú podrobne popísané v programátorskej dokumentácii.

Kapitola 4
Užívateľská dokumentácia

Táto aplikácia je určená na triedenie elektronickej pošty do rôznych kategórií. Na začiatku budete musieť poštu triediť ručne, potom to bude robiť program za Vás.

4.1 Inštalácia

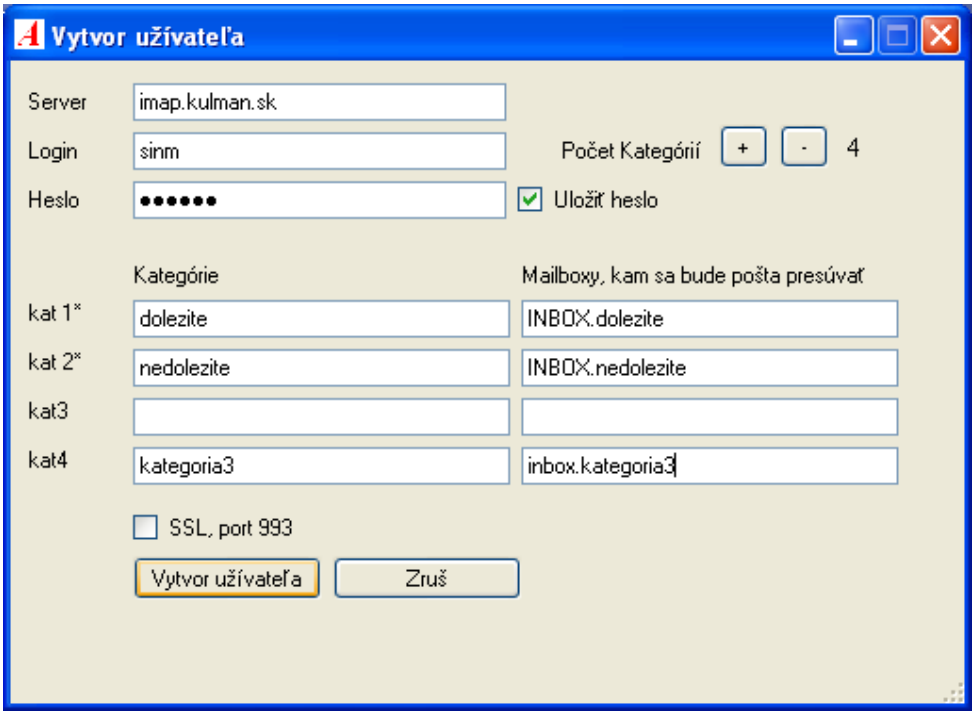
Program AClassifier je napísaný pre Windows. O inštaláciu sa postará súbor Setup.exe. Užívateľ zvolí miesto inštalácie, na ploche sa automaticky vytvorí ikona. Nakoľko bola použitá len trial verzia knižnice Chilkat.net bude aplikácia funkčná len po dobu 30 dní a to len za predpokladu že na danom počítači už táto knižnica použitá nebola.

4.2 Spúšťanie programu

Program sa spúšťa súborom bakalarka.exe, alebo ikonou z plochy. Po prvom spustení si používateľ môže vytvoriť účet. Potom je nutné aplikáciu učiť. To je možné buď mechanicky, alebo automaticky. Oba postupy budú podrobne popísané nižšie. Po učení bude program sám schopný triediť vašu poštu.

4.3 Ovládanie programu

4.3.1 Vytvorenie používateľa

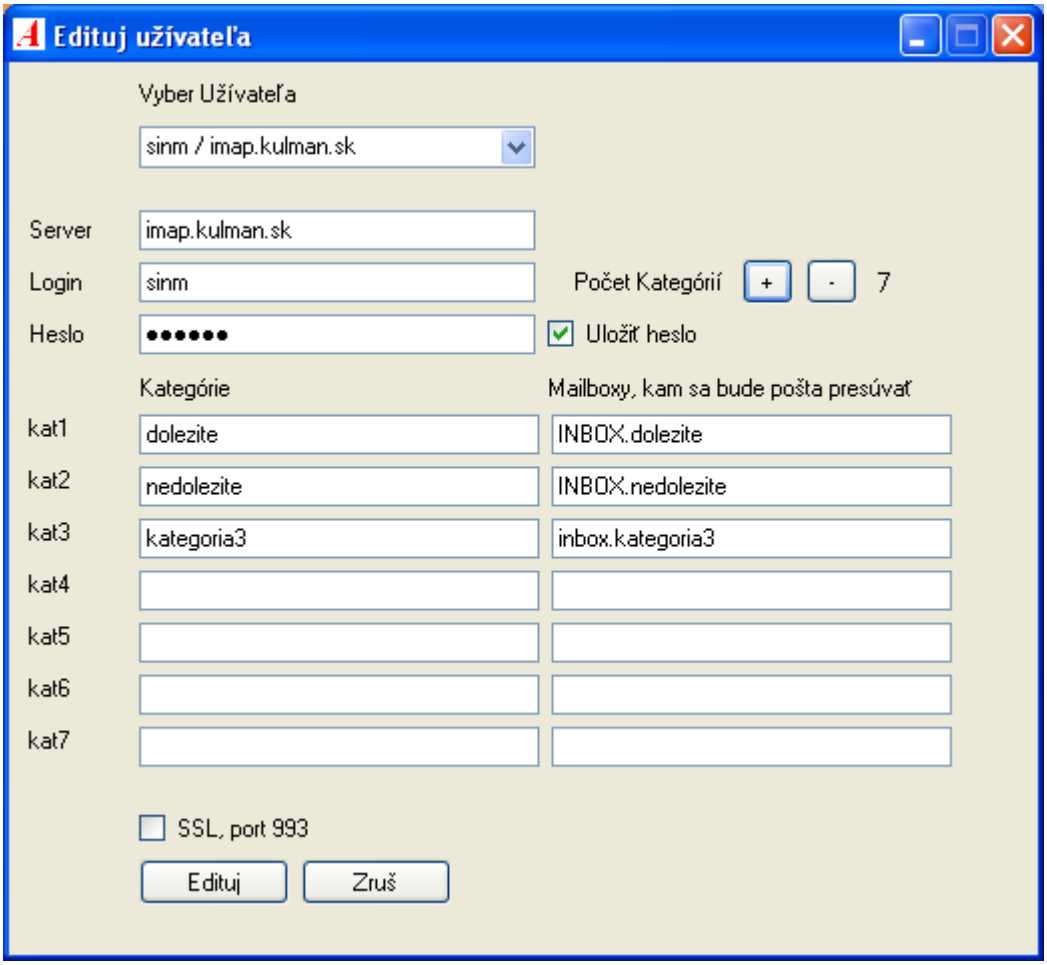


Obrázok 4.1 Vytváranie Používateľa

Na začiatku je potrebné vytvoriť prvého používateľa. V menu zvoľte možnosť Užívateľ a potom Vytvor Užívateľa. Otvorí sa Vám dialóg Vytvor Užívateľa.

Tam musíte vyplniť Váš IMAP server, login a heslo. Ďalej je nutné vyplniť minimálne 2 kategórie, do ktorých chcete svoju poštu triediť a ku každej kategórii musíte zadať názov priečinku na Vašom serveri, do ktorého chcete aby emaily danej kategórie boli presunuté. Môžete zadať aj neexistujúci priečinok. V takom prípade Vám ho program vytvorí. Niektoré servery nepovoľujú tvorbu priečinkov bez predpony INBOX., preto voľte radšej názvy priečinkov s touto predponou. Ak potrebujete poštu triediť na viac ako 2 kategórie, môžete si ich pridať, alebo odobrať tlačítkami + a -. Ak Vaša schránka vyžaduje pripojenie pomocou Secure Socket Layer (SSL) pomocou portu 993 (napríklad Gmail), zašktnite možnosť SSL, port 993. Nakoniec kliknite na tlačítko Vytvor užívateľa. Ak niektoré z povinných údajov nevyplníte, zobrazí sa Vám chybové hlásenie, ktoré Vás na to upozorní. Vytváranie používateľa zobrazuje obrázok 4.1.

4.3.2 Editovanie používateľa



Obrázok 4.2 Editovanie Používateľa

Každého používateľa je možné editovať. V menu zvoľte možnosť Užívateľ a potom Edituj Užívateľa. Otvorí sa Vám dialóg Edituj Užívateľa.

Najprv je nutné zvoliť užívateľa, ktorého chcete editovať. Následne sa Vám ukážu aktuálne informácie o zvolenom používateľovi, ktoré môžete meniť. Kategórie môžete pridávať a odoberať pomocou tlačítok + a -.

Ak zmeníte iba prihlasovacie údaje, prípadne priečinky na servery, tak doposiaľ naučené informácie budú uchované. Ak zmeníte počet kategórií, alebo názov nejakej kategórie, tak doposiaľ naučené informácie budú vynulované. Editovanie používateľa zobrazuje obrázok 4.2.

4.3.3 Mazanie používateľa

Ak chcete zmazať používateľa, v menu zvoľte možnosť Užívateľ a potom Zrušiť Užívateľa. Otvorí sa vám dialóg Zrušiť Užívateľa. V ňom zvoľte používateľa, ktorého chcete zmazať a stlačte Potvrď. Mazanie zobrazuje obrázok 4.3.

Obrázok 4.3 Mazanie používateľa

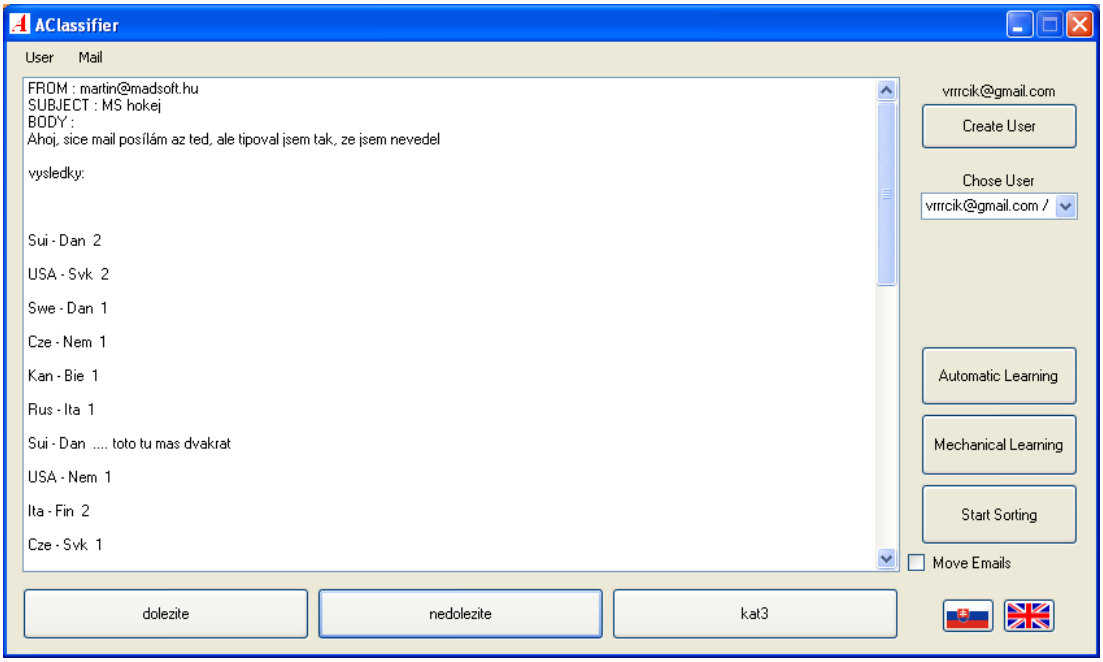
4.3.4 Jazyky

Program pozná dva jazyky a to Slovenčinu a Angličtinu. Jazyky sa dajú prepínať v hlavnom formulári tlačítkami s slovenskou a anglickou vlajkou.

4.4 Mechanické Učenie

Mechanické učenie je mód aplikácie, v ktorom budete ručne označovať, ktorá správa patrí do ktorej kategórie. Spúšťa sa z hlavného formulára. Najprv je potrebné vybrať používateľa. Učenie môžete spustiť buď tlačítkom Mechanické učenie, alebo v menu zvolíte možnosť Pošta a Mechanické učenie. V prípade, že spúšťate učenie prvý krát, tak sa Vám zobrazí okno, do ktorého musíte zadať, že koľko správ z Vašej starej pošty sa má použiť na ušenie. V každom ďalšom učení budete triediť poštu ktorá prišla po poslednej spracovanej správe. Pri triedení pošty sa Vám v spodnej časti aplikácie objavia tlačítka s názvami Vami zvolených kategórií. Pre každú správu, ktorá sa Vám zobrazí, musíte kliknúť na tlačítko s názvom kategórie, do

ktorej patrí. Učenie môžete prerušiť kedykoľvek zatvorením aplikácie. Mechanické učenie zobrazuje obrázok 4.4.



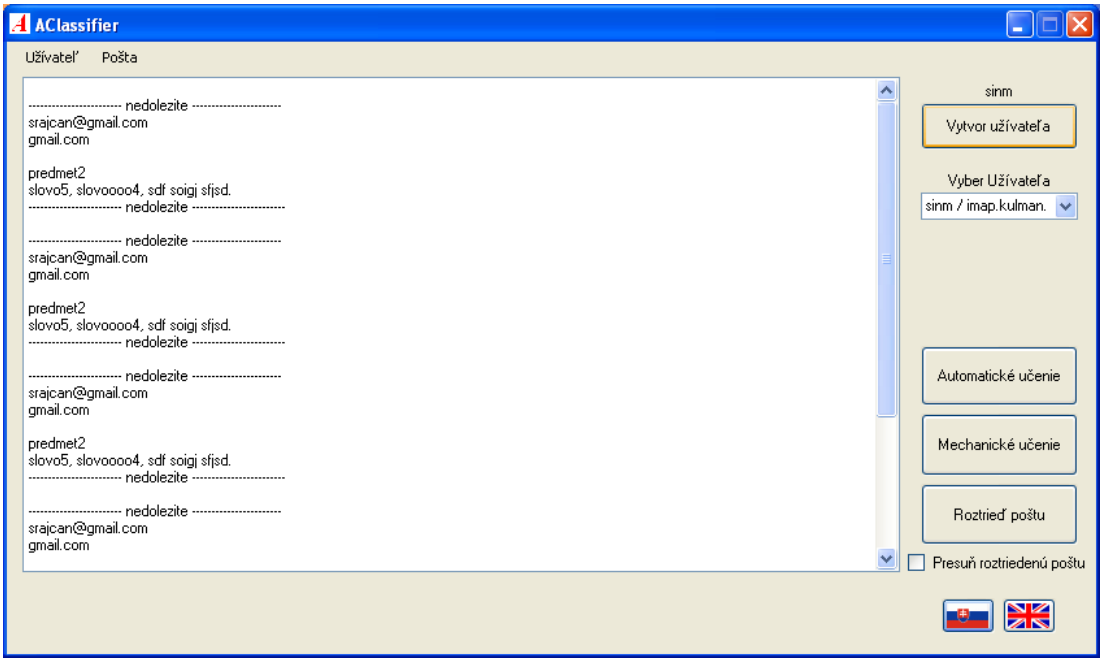
Obrázok 4.4 Učenie

4.5 Automatické učenie

Automatické učenie je mód aplikácie, v ktorom program sám zistí, že do ktorých kategórií pošta patrí na základe akcií užívateľa v jeho poštovom klientovi. Používateľ najprv vo svojom poštovom klientovi presunie vybrané správy do priečinkov, ktoré prináležia uvažovaným kategóriám. Potom v programe, v hlavnom menu spustí automatické učenie. Program potom stiahne správy z priečinkov a uloží si z nich informácie tak isto ako pri mechanickom učení. Pri prvom spustení sa Vás program opýta že koľko správ sa má sťahovať z každého priečinku. V každom ďalšom učení budete triediť poštu ktorá prišla po posledných spracovaných správach v každom priečinku.

4.6 Triedenie

Triedenie je mód aplikácie, v ktorom aplikácia sama roztriedi Vašu poštu, prípadne ju aj rozdelí do Vami zvolených priečinkov (musíte zašktnúť možnosť Presuň roztriedenú poštu). Triedenie môžete spustiť Tlačítkom Roztrieď poštu, alebo v menu možnosť Pošta / roztrieď. Triedenie zobrazuje obrázok 4.4.



Obrázok 4.4 Triedenie

Kapitola 5

Programátorská dokumentácia

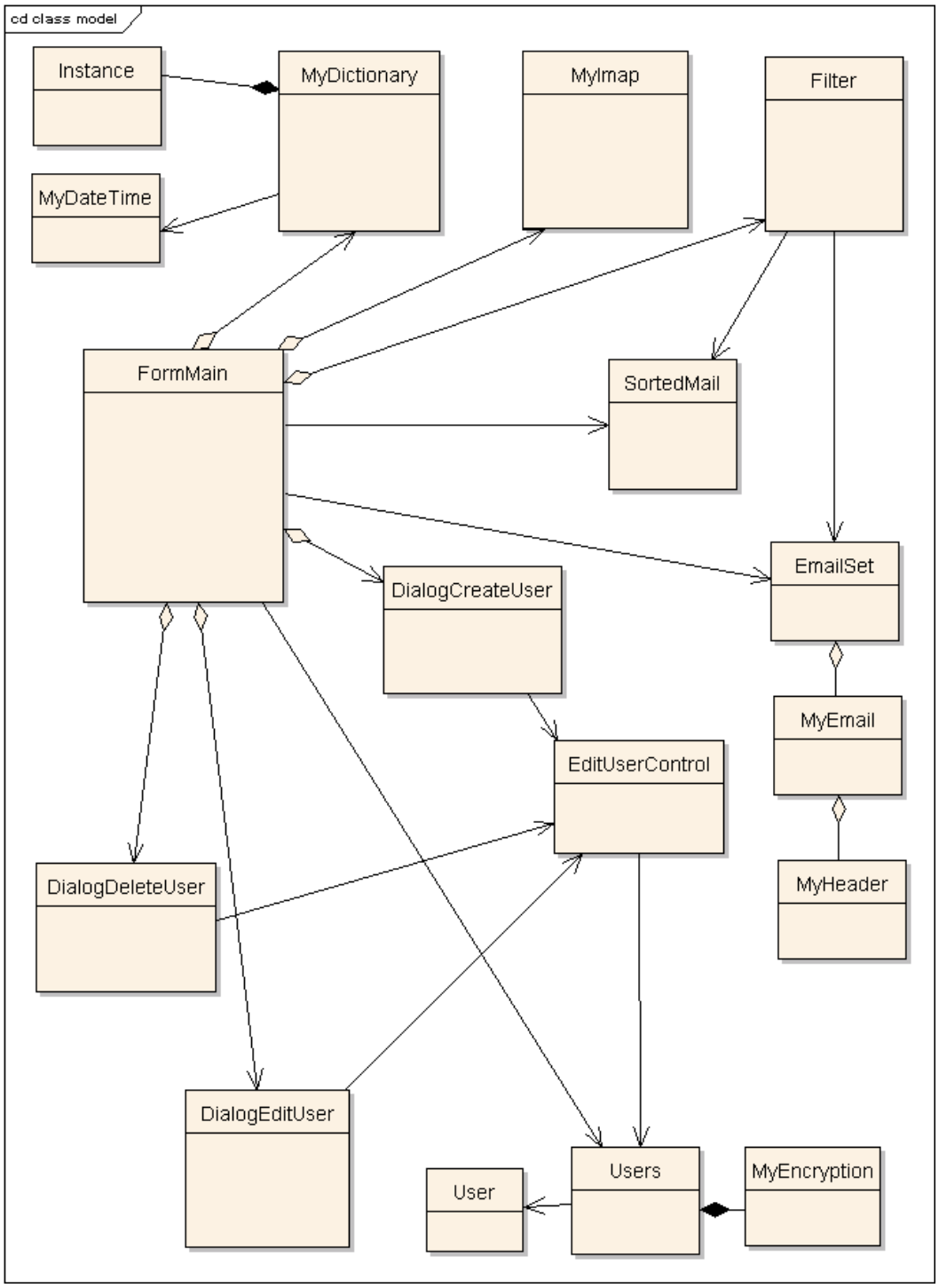
5.1 Štruktúra tried

Štruktúru tried popisuje obrázok 5.1.

5.2 Popis tried

- FormMain: hlavný formulá,r z ktorého sa celý program ovláda
- MyDictionary: trieda, ktorá spravuje databázu informácií získaných z učenia
- MyImap: trieda, ktorá je zodpovedná za komunikáciu s knižnicou Chilkat IMAP
- Filter: trieda, ktorá je zodpovedná za triedenie správ
- SortedMail: trieda, ktorá reprezentuje utriedenú správu
- EmailSet: trieda, ktorá spravuje zoznam správ
- MyEmail: trieda, ktorá reprezentuje správu
- MyHeader: trieda, ktorá reprezentuje hlavičku správy
- Users: trieda, ktorá spravuje databázu informácií o používateľoch
- User: trieda, ktorá reprezentuje jedného používateľa
- DialogCreateUser: formulár, z ktorého sa vytvárajú nový používatelia
- DialogEditUser: formulár, z ktorého sa uditujú používatelia
- DialogDeleteUser: formulár, z ktorého sa mažú používatelia
- EditUserControl: pomocná trieda, ktorú využívajú dialogy správy používateľov
- Instance: trieda, ktorá reprezentuje zoznam kategórií

- MyDateTime: statická trieda, ktorá vracia dátum v tvare dd-MMM-yyyy
- MyEncryption: statická trieda, ktorá sa stará o šifrovanie hesiel



Obrázok 5.1 Štruktúra tried

5.3 Správa dát

O správu dát získaných z učenia a triedenia sa stará trieda MyDictionary.

Všetky slová z tiel, adries, domén, predmetov a mailerov správ sú ukladané v triede SortedDictionary, pretože pri práci s naučenými dátami je potrebný rýchly prístup k informáciám o každom slove a možnosť prejsť všetky naučené slová a uložiť ich do databázy. Kľúče sú samotné slová a hodnoty sú inštancie triedy Instance v ktorej sú uložené početnosti daného slova pre každú kategóriu. Celkový počet slov pre každú zložku správy a pre každú kategóriu sú uložené v triedach List<Long>, podobne ako celkový počet správ pre každú kategóriu. V triede List<int> sú uložené najväčšie UID spracovaných správ pre priečinky kategórií, ktoré sú potrebne pri novom sťahovaní správ. UID pre priečinok Inbox je uložené v maxUIDInbox.

Trieda MyDictionary sa stará tiež o ukladanie a opätovné načítanie týchto dát z xml súborov. Informácie o telách správ a niektoré všeobecné informácie sa ukladajú do súboru „Názov servera“_“Názov účtu“_Body_“Deň založenia účtu“.xml a informácie získané z hlavičiek správ sa ukladajú do súboru „Názov servera“_“Názov účtu“_Header_“Deň založenia účtu“.xml.

O informácie o užívateľoch sa stará trieda Users. Tieto informácie sú uložené v triede List<User>. User je trieda v ktorej sú uložené prihlasovacie údaje používateľa, názov xml súborov, kde sa ukladajú jeho naučené informácie, zoznam a počet kategórií a zoznam priečinkov, kam sa má jeho roztriedená pošta ukladať. Trieda Users je taktiež zodpovedná za ukladanie a opätovné načítanie týchto informácií z xml súboru Users.xml.

5.4 Dôležité súbory

Súbory, ktoré program využíva sú nasledovné:

- Users.xml. Sú v ňom uložené informácie o každom používateľovi, ktoré boli získané pri založení účtu

```
<?xml version="1.0"?>
```

```
<users>
     <user>
     <server>názov imap serveru</server>
     <login>Váš login</login>
     <password>Vaše heslo</password>
     <bodyFileName>názov súboru kde sa nachádzajú informacie
získané z tiel správ</bodyFileName>
     <headerFileName>názov súboru kde sa nachádzajú informacie
získané z hlavičiek správ</headerFileName>
<kategorie pocet="2">
     <kategoria>dolezite</kategoria>
     <kategoria>nedolezite</kategoria>
</kategorie>
<mailboxy pocet="2">
     <mailbox>inbox.dolezite</mailbox>
     <mailbox>inbox.nedolezite</mailbox>
</mailboxy>
     <gmail>True</gmail>
</user>
</users>
```

- „Názov servera“_“Názov účtu“_Body_“Deň založenia účtu“.xml Sú do neho ukladané informácie o danom používateľovi získané pri učení z tiel správ. Taktiež sú tam informácie o tom, že odkedy sa majú sťahovať ďalšie správy pri učení alebo triedení. Tu je príklad takého súboru:

```
<?xml version="1.0"?>
<words>
 <slovo kategorie0="13" kategorie1="0" kategorie2="0"
 kategorie3="0"></slovo>
 <slovo kategorie0="0" kategorie1="0" kategorie2="0"
 kategorie3="1">zvolili</slovo>
 <slovo kategorie0="0" kategorie1="6" kategorie2="0"
 kategorie3="0">zvolite</slovo>
 <pocetSlovKat0>14077</pocetSlovKat0>
 <pocetSlovKat1>10877</pocetSlovKat1>
 <pocetSlovKat2>973</pocetSlovKat2>
 <pocetSlovKat3>6957</pocetSlovKat3>
 <pocetEmailovKat0>67</pocetEmailovKat0>
 <pocetEmailovKat1>33</pocetEmailovKat1>
 <pocetEmailovKat2>14</pocetEmailovKat2>
 <pocetEmailovKat3>14</pocetEmailovKat3>
 <since>19-May-2009</since>
 <dateTime>4/23/2009 7:14:44 PM</dateTime>
<uidCategories k0="0" k1="0" k2="0" k3="0"  />
      <uidINBOX>560</uidINBOX>
</words>
```

- „Názov servera“_“Názov účtu“_Header_“Deň založenia účtu“.xml. Sú do neho ukladané informácie o danom používateľovi získané pri učení z hlavičiek správ. Tu je príklad takého súboru:

```
<?xml version="1.0"?>
<headers>
 <address kategorie0="9" kategorie1="0" kategorie2="0"
 kategorie3="0">srajcan@gmail.com</address>
 <domain kategorie0="6" kategorie1="4" kategorie2="0"
 kategorie3="0">gmail.com</domain>
 <subject kategorie0="0" kategorie1="2" kategorie2="0"
 kategorie3="0">prikladsubject</subject>
 <mailer kategorie0="0" kategorie1="0" kategorie2="5"
 kategorie3="0">microsoft outlook express 5.50.4522.1200</mailer>
 <otherInformations>
  <pocetSlovAddressKat0>67</pocetSlovAddressKat0>
  <pocetSlovAddressKat1>33</pocetSlovAddressKat1>
  <pocetSlovAddressKat2>14</pocetSlovAddressKat2>
  <pocetSlovAddressKat3>14</pocetSlovAddressKat3>
  <pocetSlovDomainKat0>67</pocetSlovDomainKat0>
  <pocetSlovDomainKat1>33</pocetSlovDomainKat1>
  <pocetSlovDomainKat2>14</pocetSlovDomainKat2>
  <pocetSlovDomainKat3>14</pocetSlovDomainKat3>
  <pocetSlovSubjectKat0>175</pocetSlovSubjectKat0>
  <pocetSlovSubjectKat1>115</pocetSlovSubjectKat1>
  <pocetSlovSubjectKat2>24</pocetSlovSubjectKat2>
  <pocetSlovSubjectKat3>56</pocetSlovSubjectKat3>
  <pocetSlovMailerKat0>67</pocetSlovMailerKat0>
  <pocetSlovMailerKat1>33</pocetSlovMailerKat1>
  <pocetSlovMailerKat2>14</pocetSlovMailerKat2>
  <pocetSlovMailerKat3>14</pocetSlovMailerKat3>
 </otherInformations>
</headers>
```

5.5 Riešenie niektorých problémov

5.5.1 Výpočet pravdepodobnosti

Ak sa vzorec na výpočet pravdepodobnosti (5) aplikuje na dlhšie správy, výsledné hodnoty V_{Class} budú veľmi malé čísla. Preto je nutné už medzivýsledky prenásobovať nejakou konštantou. Ak medzivýsledok produktu vo vzorci (5) klesne pod hranicu 10^-10, všetky medzivýsledky budú prenásobené číslom 10^10.

5.5.2 Problém s formátom dátumu

Ukázalo sa, že metóda DateTime.now.ToString("dd-MMM-yyyy"); dáva na rôznych počítačoch rôzne výsledky, preto bola vytvorená trieda MyDateTime, ktorá tento problém rieši.

5.5.3 Problémy s pripojením na server

Pri pripojení na server občas dochádza k chybám, a preto sú všetky funkcie, ktoré pracujú so serverom, v try blokoch.

5.5.4 Šifrovanie hesiel

Heslá sa ukladajú zašifrované pomocou algoritmu DES. Za šifrovanie a dešifrovanie hesiel je zodpovedná trieda MyEncryption.

5.5.5 Sťahovanie správ zo serveru

Pôvodne sa správy zo serveru sťahovali podľa času posledného sťahovania. Ukázalo sa že takýto spôsob sťahovania je nesprávny, pretože program niektoré správy ignoroval a naopak niektoré (napríklad také čo mali nastavený chybný čas odosielania) sa brali do úvahy viackrát. Preto sa teraz správy sťahujú na základe ich UID (unikátne číslo, ktoré im priradí server pri zaradení správy do priečinku). Správa ktorá bola do priečinku zaradená neskôr má väčšie UID ako správa, ktorá tam bola zaradená skôr. Pre každý priečinok si aplikácia pamätá UID poslednej spracovanej správy a pri ďalšej synchronizácií sa berú do úvahy len správy s vyšším UID.

Napriek tomu že sa informácie o poslednom sťahovaní správ už nevyužívajú sa v programe aj naďalej uchovávajú pretože by mohli byť užitočné pri jeho ďalšom rozširovaní.

5.5.6 Názvy priečinkov

Priečinok na imapovom serveri v ktorom je doručená pošta má podľa [9] názov Inbox a nieje Case-sensitive. Názvy ostatných priečinkov môžu a nemusia byť Case-sensitive, záleží to na konkrétnej implementácii serveru. Preto sa pri zakladaní a editovaní užívateľa postupuje nasledovne :

- Ak na serveri zadaný priečinok case-sensitive existuje, program bude pracovať s týmto priečinkom
- Ak na serveri zadaný priečinok case-unsensitive existuje, program bude pracovať s týmto priečinkom

- Ak zadaný priečinok case-unsensitive na serveri neexistuje, program sa ho pokúsi vytvoriť
- Ak sa priečinok podarí vytvoriť, program bude pracovať s týmto priečinkom
- Ak sa priečinok s daným názvom nepodarí vytvoriť a na servery priečinok s rovnakým názvom case-unsensitive neexistuje, užívateľ na to bude upozornený a program si zapamätá že má pracovať s neexistujúcim priečinkom a používateľa na to bude pri učení a triedení upozorňovať. Užívateľ neskôr môže tento priečinok vytvoriť vo svojom poštovom klientovi.

Popísaný algoritmus je implementovaný v triedach EditUserControl, MyImap a Users.

5.6 Testovanie úspešnosti algoritmu

Prvý test bol robený na schránke srajcan@gmail.com. Pošta sa triedila do štyroch kategórií. Na učenie bolo použitých 103 správ. Program potom sám roztriedil 164 správ, z toho 117 správne, čo je 71%.

Druhý test bol robený na schránke regi77@gmail.com. Pošta sa triedila do štyroch kategórií. Na učenie bolo použitých 60 správ. Program potom sám roztriedil 100 správ, z toho 75 správne, čo je 75%.

Kapitola 6

Záver

Cieľom tejto bakalárskej práce bolo vytvoriť program, ktorý by umožnil adaptívnu klasifikáciu došlej elektronickej pošty. V rámci práce vznikol program Aclassifier, ktorý to umožňuje.

Nakoľko existuje množstvo programov na detekciu spamu, tak snahou tejto práce bolo zamerať sa na klasifikáciu pozitívnej, teda želanej pošty.

Program je interaktívna aplikácia pracujúca pod systémom Windows.

Program na spojenie s poštovom serverom používa protokol IMAP, ktorý umožňuje plnohodnotnú prácu zo schránkou.

Program pracuje na princípe Bayesovej teórie, ktorá sa mnoho krát osvedčila pri detekcii spamu.

Program si pamätá údaje o pošte každého používateľa zvlášť, takže ho môžu využívať viacerí používatelia jedného počítača.

Na rozsiahle testovanie nebol čas, pretože program je potrebné testovať v reálnom čase aspoň niekoľko mesiacov, ale pri 100 správach použitých na učenie, program triedi ostatné správy s úspešnosťou okolo 75%.

6.1 Ďalší rozvoj programu

V budúcnosti by sa program mohol rozvíjať viacerými smermi.

- Vytvoriť klasifikátor, ktorý by došlú poštu klasifikoval na 100 % správne, je zrejme nemožné a vytvoriť klasifikátor, ktorý by sa k tejto hranici aspoň blížil, je pre jedného študenta veľmi náročné, preto je možné vytvorený triediaci algoritmus ešte zdokonaľovať

- Vytvoriť plnohodnotného poštového klienta. Táto možnosť je však nepravdepodobná, pretože by bolo veľmi náročné implementovať množstvo funkcií, ktoré majú externé poštové klienty.
- Vytvoriť plugin do nejakého externého poštového klienta. Tým by ale program stratil na obecnosti.
- Vytvoriť lepšie zobrazovanie správ
- Vytvoriť klasifikátor ako server bez grafického prostredia

Zoznam použitej literatury

[1] http://cs.wikipedia.org/wiki/Spam#Filtrace_podle_zp.C5.AFsobu_dopravy

[2] http://en.wikipedia.org/wiki/Bayesian_spam_filtering

[3] Raju Shrestha and Yaping Lin, Improved Bayesian Spam Filtering Based on Co-weighted Multi-area Information 2005

[4] http://www.example-code.com/csharp/imap.asp

[5] http://xemail-net.sourceforge.net/

[6] http://www.codeproject.com/KB/IP/imaplibrary.aspx

[7] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam and S. Slattery: Learning to Extract Symbolic Knowledge from the World Wide Web. In Proceedings of the 15th National Conference on Artificial Intelligence 1998

[8] Andrew McCallum and Kamal Nigam: A Comparison of Event Models for Naive Bayes Text Classification. Working notes of the 1998 AAAl/ICML workshop on Learning for Text Categorization

[9] RFC 3501 INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1