VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

KLASIFIKACE DOKUMENTŮ POMOCÍ ANALÝZY OBSAHU

DOCUMENT CLASSIFICATION USING CONTENT ANALYSIS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **FILIP BORČÍK**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. LENKA TŘEŠTÍKOVÁ**
SUPERVISOR

BRNO 2019

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav počítačové grafiky a multimédií (UPGM) Akademický rok 2018/2019

Zadání bakalářské práce

Student: **Borčík Filip**
Program: Informační technologie
Název: **Klasifikace dokumentů pomocí analýzy obsahu**
Content Document Classification
Kategorie: Bezpečnost

Zadání:

1. Seznamte se se standardy ISO 27000, prostudujte principy pro zpracování textu pomocí analýzy obsahu na základě definovaných pravidel a analyzujte způsob, jak automaticky klasifikovat dokumenty.
2. Na základě analýzy navrhněte architekturu pro automatickou klasifikaci dokumentů v prostředí MS Office.
3. Implementujte navržený systém složený z doplňku v MS Office, klasifikační služby a databáze.
4. Nástroj otestujte na vhodném testovacím vzorku dat. Proveďte výkonnostní testy.
5. Diskutujte možnosti dalšího rozšíření.

Literatura:

- Řada norem ISO/IEC 27000
- Dle doporučení vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování práce viz http://www.fit.vutbr.cz/info/szz/

Vedoucí práce: **Třeštíková Lenka, Ing.**
Vedoucí ústavu: Černocký Jan, doc. Dr. Ing.
Datum zadání: 1. listopadu 2018
Datum odevzdání: 15. května 2019
Datum schválení: 12. listopadu 2018

Abstrakt

Táto práca sa zaoberá klasifikáciou dokumentov podľa rodiny štandardov ISO/IEC 27000. Poukazuje na potrebu, ale aj problémy klasifikovania v korporátnom prostredí. Práca taktiež implementuje systém pre klasifikáciu dokumentov prostredia MS Office založenej na analýze obsahu pomocou definovaných pravidiel. Tento systém je zavedený do aplikácie DocTag vyvíjanej spoločnosťou AEC.

Abstract

This work deals with document classification based on standard family ISO/IEC 27000. Points to a need, but also issues of classification in corporate environment. The work also implements system for MS office documents classification based on content analysis using defined rules. This system is introduced into DocTag application developed by AEC company.

Kľúčové slová

klasifikácia dokumentov, ISO/IEC 27000, ISO/IEC 27001, MS Office, OpenXML, bezpečnosť, analýza obsahu, DocTag

Keywords

document classification, ISO/IEC 27000, ISO/IEC 27001, MS Office, OpenXML, security, content analysis, DocTag

Citácia

BORČÍK, Filip. *Klasifikace dokumentů pomocí analýzy obsahu.* Brno, 2019. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Lenka Třeštíková

Klasifikace dokumentů pomocí analýzy obsahu

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pani Ing. Lenky Třeštíkovej. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Filip Borčík
11. mája 2019

Poďakovanie

Ďakujem vedúcej bakalárskej práce, pani Ing. Lenke Třeštíkovej, za cenné rady a pomoc. Taktiež ďakujem kolegom zo spoločnosti AEC za usmerňovanie pri tvorení tejto práce.

Obsah

Kapitola 1

Úvod

V rozmanitosti mnohých organizácii, v ktorých každá má iné ciele a zaoberá sa inými záležitosťami, predsa existuje niečo, čo majú všetky spoločné. Je to vôľa uchovávať svoje údaje v bezpečí. Už od útleho veku sa učíme zachovávať tajomstvá a citlivé informácie iba v kruhu osôb, ktoré o nich vedieť môžu. Rovnako je tomu aj v korporátnom prostredí, no tu únik informácií často prináša omnoho väčšie škody.

V minulosti malo slovo osoby veľkú cenu. Keď niekto vyzradil niečo, o čom povedal, že to uchová v tajnosti, bol často odsudzovaný spoločnosťou a musel z tohto činu vyvodiť dôsledky. K získaniu informácií preto bolo zväčša nutné použiť násilie v podobe múk. Dnešná doba je plná nedôvery a podrazov. V spojení s digitalizáciou značne uľahčuje proces odcudzenia informácií.

Dennodenne sa po celom svete vytvárajú milióny nových dokumentov. Banky behom jedného dňa uzatvárajú s klientmi tisíce nových zmlúv. V zdravotníctve sa vytvorí nespočetné množstvo nových správ o diagnózach pacientov. Všetko sú to dokumenty, ktoré je nutné chrániť pred ich zneužitím. Väčšina obsahuje osobné údaje, no množstvo dát je plné informácií, ktoré vedú k obrovským finančným ziskom. Snaha o získanie takýchto dokumentov je čoraz väčšia a je stále ťažšie zabrániť novým útokom.

Cieľom tejto práce je pozrieť sa na jedno z riešení ochrany dát, ktorým je klasifikácia dokumentov. Po tomto kroku môžeme kontrolovať a obmedzovať prístup ku dôverným dátam. Okrem teoretických znalostí, založených na poznatkoch z medzinárodných noriem, prináša práca aj implementáciu rozpoznávania klasifikačnej triedy dokumentov analýzou ich obsahu.

Kapitola 2 opisuje klasifikáciu dokumentov z pohľadu štandardov ISO/IEC 27000. Ukazuje niekoľko nástrojov, ktoré sa v súčastnosti na klasifikáciu využívajú. Popisuje taktiež aplikáciu *DocTag*, ktorá je neskôr použitá pri vytváraní výsledného systému.

V ďalšej časti je možné nahliadnuť do návrhu a postupu pri implementácii odovzdaného riešenia. Ukážeme si, ako vyzerá jazyk pre zápis pravidiel klasifikačných tried, ale aj ako extrahovať a analyzovať text z dokumentov balíku Microsoft Office. Medzi podporované boli vybrané aplikácie Word, Excel, PowerPoint a Visio. Toto riešenie bolo vytvorené v spolupráci so spoločnosťou AEC.

Nezabudlo sa ani na testovanie riešenia. Pri testovaní sme sa zamerali najmä na rýchlosť využitých metód, avšak nechýba ani test validity klasifikácie. Riešenie je taktiež možné rozšíriť o niekoľko ďalších vlastností, ktorých podrobnejší popis nájdeme na konci práce.

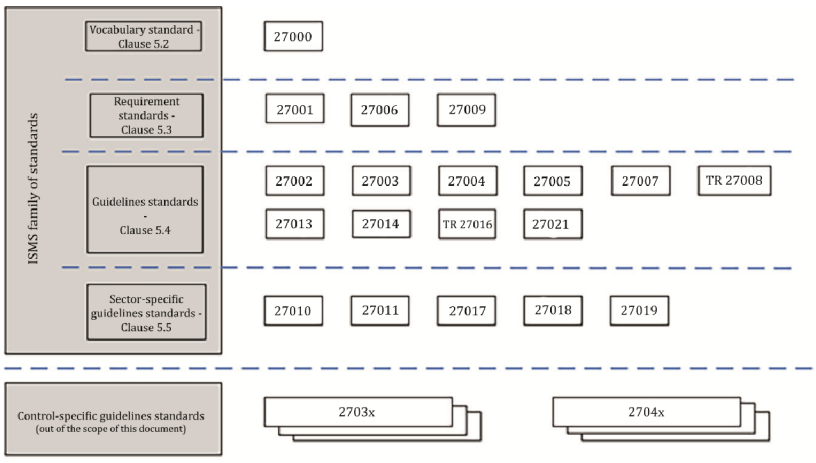
Kapitola 2

Analýza

V tejto kapitole je možné nájsť teoretické priblíženie problematiky klasifikácie dokumentov. Okrem iného si rozoberieme problémy pri klasifikovaní a pozrieme sa aj na niektoré nástroje, ktoré nám s klasifikáciou pomáhajú.

2.1 Normy a štandardy

Pri vytváraní bezpečného informačného prostredia organizáciám pomáha rodina štandardov ISO/IEC 27000. Ochrana finančných informácií, podrobností o zamestnancoch alebo informácií o zákazníkoch a tretích stranách, je s ich pomocou hravo zvládnuteľná. Normy ISO 27000 zahŕňajú mnohé osvedčené postupy, ktoré boli vyvíjané počas celých generácií. Poskytujú taktiež praktické nástroje pre systém manažmentu informačnej bezpečnosti (Information Security Management System - ISMS). Z obrovského výberu noriem, ktoré sú definované v tejto rodine štandardov, si jednotlivé organizácie musia vybrať tú, ktorá im pomôže čo najlepšie vyriešiť ich aktuálne problémy. Členenie noriem ISO/IEC je znázornené na obrázku 2.1 [12].



Obr. 2.1: Štandardy rodiny ISMS, [12]

- **ISO/IEC 27000** – vysvetlenie terminológie používanej v ISO/IEC 27001 [12]

Štandardy popisujúce požiadavky [12]:

- **ISO/IEC 27001** – hlavná norma systému manažmentu informačnej bezpečnosti
- **ISO/IEC 27006** – požiadavky na orgány poskytujúce audit a certifikáciu systémov manažmentu informačnej bezpečnosti
- **ISO/IEC 27009** – požiadavky na použite ISO/IEC 27001 v rôznych špecifických odvetviach

Štandardy popisujúce všeobecné usmernenia [12]:

- **ISO/IEC 27002** – usmernenie implementácie kontroly informačnej bezpečnosti
- **ISO/IEC 27003** – usmernenie implementácie ISMS
- **ISO/IEC 27004** – monitorovanie, meranie, analýza a hodnotenie ISMS
- **ISO/IEC 27005** – manažment rizikovosti informačnej bezpečnosti
- **ISO/IEC 27007** – pokyny pre audit ISMS
- **ISO/IEC 27008** – pokyny pre audítorov kontroly informačnej bezpečnosti
- **ISO/IEC 27013** – usmernenie k integrovanej implementácii ISO/IEC 27001 a ISO/IEC20000-1
- **ISO/IEC 27014** – usmernenie princípov a procesov riadenia informačnej bezpečnosti
- **ISO/IEC 27016** – ekonomika informačnej bezpečnosti
- **ISO/IEC 27021** – požiadavky na spôsobilosť odborníkov ISMS

ISO/IEC 27001

ISO 27001 je medzinárodná norma starajúca sa o riadenie rizík pre čo najlepšie zabezpečenie informácií. Pomocou prístupu založeného na procesoch zabezpečuje vytvorenie, implementáciu, prevádzku, monitorovanie a udržiavanie systému riadenia informačnej bezpečnosti [11]. Tento certifikát je vhodný pre každého kto pracuje s informáciami, nech sa už jedná o IT služby, štátnu správu, telekomunikačných operátorov a iné. Od svojho publikovania v októbri 2005 sa považuje za nástupcu britského štandardu BS7799. Neberie ohľad na typ organizácie, vďaka čomu môže byť implementovaná v organizáciách ziskových aj neziskových, veľkých aj malých, patriacich do súkromného ale aj verejného sektoru [18]. Podporuje súlad s mnohými zákonmi vrátane európskeho nariadenia o ochrane osobných údajov (General Data Protection Regulation - GDPR) a smernice o bezpečnosti sietí a informačných systémov (Network and Information Systems - NIS). Zavádza princípy Organizácie pre hospodársku spoluprácu a rozvoj (Organisation for Economic Co-operation and Development - OECD) pre oblasť bezpečnosti informačných systémov a sietí. Vďaka týmto vlastnostiam sa táto norma zaradila medzi celosvetovo najpopulárnejšie normy týkajúce sa informačnej bezpečnosti [18, 11].

ISO/IEC 27001 nezaväzuje používateľa k špecifickým kontrolám, pretože organizácie, ktoré prijali tento štandard majú rôzny prístup k požadovaným kontrolám. Obsahom tejto normy sú bezpečnostné požiadavky bližšie definované v ISO/IEC 27002, a preto si môžu organizácie vyberať akékoľvek kontroly informačnej bezpečnosti na základe ich konkrétneho rizika [11, 12]. Kľúčom k výberu príslušných kontrol je jedna z najdôležitejších častí ISMS a to vykonávanie úplného hodnotenia informačného rizika organizácie. Vedenie má pri výbere kontrol bez použitia ISMS možnosť pristúpiť k jednotlivým rizikám podľa svojho vlastného uváženia. Môže sa im vyhnúť, zdieľať ich, prípadne ich môže akceptovať. Avšak, pri používaní ISMS táto vlastnosť zaniká a po vyhodnotení rizík je nutné implementovať všetky kontroly, ktoré vychádzajú z posúdenia rizík. Pokiaľ chceme dosiahnuť najvyššiu úroveň ochrany, je potrebné aplikovať prevenciu v čo najväčšej miere. Je nutné mať jasne stanovené niekoľko základných položiek, medzi ktoré patria politiky a postupy, právna ochrana, disciplinárne opatrenia, odborná príprava a povedomie osôb, ktorí sa môžu dostať do styku s informáciami a iné [11, 16, 18].

Je potrebné dávať pozor na to aby tieto opatrenia boli iniciatívou vyššieho manažmentu spoločnosti. Jedine tlakom z najvyšších miest je možné dosiahnuť rozšírenie ISMS v celej štruktúre organizácie a dostať tieto politiky medzi všetkých zamestnancov. ISO 27001 myslí aj na túto potrebu a podáva zoznam požiadaviek na vyšší manažment organizácie [16]:

- nastaviť ciele na informačnú bezpečnosť,
- vytvoriť a publikovať politiky pre kontrolu splnenie týchto cieľov,
- stanoviť zodpovednosť za informačnú bezpečnosť,
- poskytnúť dostatok zdrojov, finančných aj ľudských,
- pravidelne kontrolovať naplnenie cieľov.

Nech robíme čo sa dá, neovplyvníme rozvoj nových technológií, či zmeny v štruktúre spoločnosti. To môže mať za následok, že kedysi dokonale vytvorená ochrana informácií bude obsahovať diery predstavujúce hrozbu. ISO 27001 káže dodržiavať rôzne metódy aby sme predišli týmto problémom. Monitorovanie a meranie, nápravné opatrenia, interné audity, či iné postupy stanovené týmto štandardom sa pravidelne upravujú, aby boli stále aktuálne a schopné zabezpečiť informačnú bezpečnosť v každej spoločnosti [16].

ISO/IEC 27002

Ako už bolo vyššie spomenuté, obsah normy ISO 27002 bližšie popisuje štandard ISO 27001. Zatiaľ čo 27001 popisuje jeden bod v jednej vete, štandard 27002 tento bod rozvíja až na dĺžku jednej strany. Táto norma je založená na už osvedčených postupoch a jej hlavným cieľom je zabezpečiť pokyny a princípy pre začatie, implementovanie, udržiavanie a zlepšovanie riadenia informačnej bezpečnosti v spoločnosti. Štruktúru normy ISO/IEC 27002 tvorí niekoľko sekcií popisujúcich základné usmernenia k vybudovaniu ISMS [19, 11]:

1. **Posudzovanie rizík**

   Organizácia by mala vyhodnotiť riziká týkajúce sa bezpečnosti informácií. Dôraz musí klásť na prioritné ciele, ktoré najviac ovplyvnia výber rizikových operácií.

2. **Politika informačnej bezpečnosti**

   Je potrebné vytvoriť dokument obsahujúci politiku informačnej bezpečnosti danej organizácie, ktorý by hovoril o tom, aká je štruktúra pre dosiahnutie stanovených cieľov. Obsahovať by mal taktiež popis formy kontrol a pokuty, ktoré budú dané v prípade nedodržania politiky. Zároveň je mimo iné nutné, aby sa manažment danej spoločnosti zaviazal k dodržiavaniu tejto politiky.

3. **Organizácia informačnej bezpečnosti**

   Aby sme mohli správne riadiť informačnú bezpečnosť v organizácii, je potrebné, aby všetky aktivity týkajúce sa tohto problému boli koordinované osobami chrániacimi informácie dôverného charakteru.

4. **Správa aktív**

   Dáta, zmluvy, užívateľské príručky, software, vývojové nástroje a iné patria medzi aktíva, ktoré majú pre organizáciu určitú hodnotu. Z toho dôvodu by mali byť klasifikované do určitých kategórií a dokumentom by mali byť vymedzené pravidlá, ktoré by vraveli ako sa s danými aktívami môže narábať.

5. **Zabezpečenie ľudských zdrojov**

   Je nesmierne dôležité vedieť s kým pracujeme. Pri výbere zamestnancov je preto nutné analyzovať každú osobu a následne ju dosadiť na príslušnú pozíciu. Cieľom tejto sekcie je minimalizovať riziko odcudzenia a zneužitia dôležitých dokumentov.

6. **Fyzická a environmentálna bezpečnosť**

   Zariadenia a dokumenty by mali byť v prostredí, ktoré im poskytne prislúchajúcu ochranu a dostupnosť. Ak sa jedná o dáta s vyššou mierou potrebnej ochrany je vhodné použiť aj trezor.

7. **Riadenie komunikácie a operácií**

   V spoločnostiach by mala byť zabezpečená ochrana informácií pomocou správneho manažmentu, ktorým by sa dosiahlo minimalizácie rizík. Ide o napríklad zálohovanie dát alebo zabezpečenie sieťovej komunikácie.

8. **Riadenie prístupu**

   Prístup a spracovanie informácií môže náležať iba autorizovaným užívateľom a nikomu inému. Takto je možné predísť poškodeniu dokumentov a zdrojov pracujúcimi s informáciami.

9. **Získanie, vývoj a údržba informačných systémov**

   Ešte pred implementáciou informačných systémov musia byť dohodnuté požiadavky na ich bezpečnosť. Zachovanie dôvernosti, autentickosti a integrity môžeme dosiahnuť aj s pomocou kryptografie.

10. **Manažment incidentov informačnej bezpečnosti**

    Pre zamestnancov, dodávateľov a tretie strany je potrebné stanoviť formálne postupy ohlasovania udalostí týkajúcich sa informačnej bezpečnosti. Čím skôr sa jednotlivé problémy nahlásia, tým skôr je ich možné odstrániť.

11. **Riadenie kontinuity podnikania**

    Vytvorenie a realizácia plánu na prevenciu proti narušeniu podnikateľských aktivít a pre zabezpečenie rýchlej obnovy jednotlivých podnikateľských operácií.

12. **Dodržiavanie**

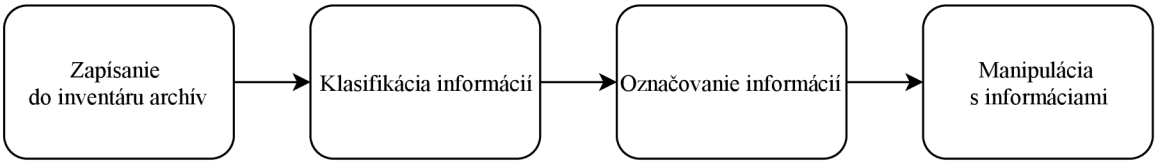
    Táto sekcia hovorí o dodržiavaní trestného a občianskeho práva. Ak chceme dosiahnuť súlad so zákonom je teda nutné pravidelne kontrolovať jeho dodržiavanie.

2.2 Klasifikácia informácií

Ako bolo už vyššie spomenuté, klasifikácia informácií hrá veľkú rolu pri ich ochrane. Správny ISMS by mal mať vytvorenú politiku, ktorá by klasifikáciu zahŕňala. Bez nej nie je možné vytvoriť správne podmienky pre dokonalú ochranu informácií.

Klasifikácia podľa ISO/IEC 27001

Proces riadenia klasifikovaných informácií je v ISO 27001 rozdelený do štyroch menších častí, ktoré môžeme vidieť na obrázku 2.2.



Obr. 2.2: Proces riadenia klasifikovaných informácií

Zapísanie do inventáru aktív

Norma v sekcii A.8.1.1 hovorí o zaradení informácií do inventáru aktív organizácie. Nie je možné klasifikovať informácie kým nevieme, že tie informácie existujú. Nech sú informácie akéhokoľvek typu, musia mať svojho vlastníka, ktorý za nich nesie zodpovednosť. Väčšinou sa jedná o osobu, ktorou boli tieto informácie vytvorené ale nemusí to tak byť vždy [15, 9].

Informácie prichádzajú v rôznych podobách. Slovne prenášané informácie, papierové a elektronické dokumenty, informačné systémy a databázy, emaily ale aj rôzne pamäťové médiá patria medzi aktíva, ktoré by mali byť zabezpečené. Z tohoto dôvodu je nutná ich klasifikácia [13].

Klasifikácia informácií

Aktíva sa klasifikujú do určitých kategórií, v ktorých každá upravuje ich dostupnosť a dôvernosť. Každá organizácia si ale musí definovať sama aké kategórie bude mať a čo v danej organizácii jednotlivé kategórie znamenajú. Je dôležité upozorniť na to, že ISO 27001 nepredpisuje kategórie klasifikovania. Pri vytváraní jednotlivých kategórií sa väčšinou očakáva, že čím väčšia je spoločnosť v ktorej sa budú informácie klasifikovať, tým viac kategórií bude mať. Avšak, toto je iba mylná domnienka a overené postupy vravia, že je lepšie mať kategórií menej. Zvyčajne sa zvykne klasifikovať do 3-5 klasifikačných tried. Najbežnejší spôsob klasifikovania obsahuje 2 dôverné triedy, 1 verejnú a vyzerá nasledovne [13]:

- **Verejné** – informácie môže vidieť každý bez obmedzenia obsahu. V tejto kategórii nenájdeme užívateľov bez prístupu. Jedná sa zväčša o rôzne marketingové a informačné materiály organizácie, jej webové stránky, či informácie z verejných zdrojov.

- **Interné** – informácie môžu byť odhalené a šírené v rámci danej organizácie bez obmedzenia obsahu alebo času. Mimo organizáciu, partnerom a iným osobám, sa môžu šíriť len s povolením zodpovednej osoby. Príkladom takýchto informácií môžu byť vnútorné predpisy organizácie.

- **Chránené** – citlivé informácie s vysokou mierou dôvernosti. Ich odcudzenie môže mať veľký dopad na chod spoločnosti. Finančné poškodenie, poškodenie dobrého mena, zrušenie zmlúv, prepad spoločnosti. Toto všetko sú príklady následkov, ktoré môžu nastať. Prístup k chráneným informáciám majú preto iba vybrané osoby, prípadne skupiny osôb. Je treba poznamenať, že medzi chránené informácie patria okrem strategických plánov, či obchodných informácií, aj osobné údaje zamestnancov.

Niektoré schémy, ako napríklad schéma vlády Spojeného kráľovstva, neobsahujú ani jednu verejnú triedu alebo naopak, ako napr. NATO, obsahujú viac verejných tried. Všetko záleží na samotnej organizácii a jej definícii klasifikačných tried. Celý tento postup je opísaný v sekcii A.8.2.1 [15, 9].

Označovanie informácií

Hneď po klasifikácii je potrebné informácie označiť. Štandard ISO/IEC 27001 v sekcii A.8.2.2 nehovorí o presnom spôsobe označovania a opäť je potrebné si v organizácii tento spôsob zadefinovať [15, 9]. Označovanie má zväčša na starosti vlastník aktíva. Je dobré označovať dôveryhodnejšie aktíva prioritne. V prípade, že nie je možné označovať každé jedno aktívum, je potrebné označiť aspoň lokáciu skupiny aktív (priečinok, miestnosť, zásuvku, ...). Zoskupovanie aktív podľa ich klasifikačnej kategórie môže značne pomôcť pri administratíve [13].

Veľkou pomocou pri práci s už označenými informáciami je dostatočná viditeľnosť klasifikačnej značky. Ak sa jedná o dokumenty, značka by mala byť umiestnená napríklad na prvej strane dokumentu alebo v hlavičke, prípadne pätičke dokumentu. Ak sa jedná o iné elektronické aktívum, je možné ho označiť v metadátach a následne pomocou rôznych nástrojov (napr. DLP – Data Loss Prevention, prevencia dátových strát pomocou monitorovania a blokovania citlivých dát) a správnych politík zabezpečiť jeho dôvernosť. Na označovanie dokumentov v súčasnosti existuje niekoľko softvérových nástrojov, ktoré sa o klasifikáciu dokážu postarať a tak uľahčia prácu užívateľom. Príkladom je aj nástroj DocTag vyvíjaný spoločnosťou AEC [13, 15].

Manipulácia s informáciami

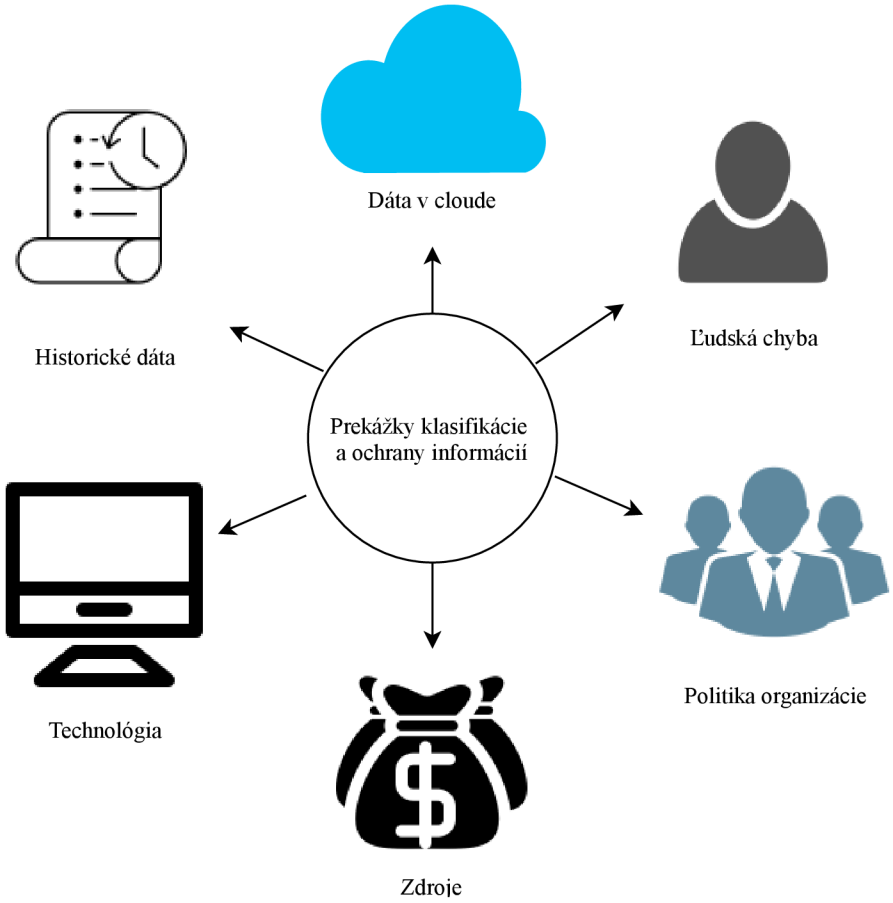
Jednou z najkomplexnejších častí riadenia informácii v rámci systému ISMS je práve zaobchádzanie s klasifikovanými informáciami. Pokiaľ budeme mať aktíva klasifikované a označené ale nebudeme s nimi zaobchádzať primerane podľa ich kategórie dôvernosti, nikdy nedosiahneme cieľa, ktorým je ich bezpečie. Sekcia A.8.2.3 normy ISO 27001 vraví o vytvorení politiky organizácie, ktorá by nedovolila pristupovať nedostatočne dôveryhodným osobám, ale samozrejme aj softvérom, k informáciám, ku ktorým nemajú mať prístup. Medzi dôveryhodnými informáciami sa často nachádzajú rámce informácií, ktorých manipulácia

spadá pod nariadenia vládnych oddelení. Môžeme spomenúť nariadenie GDPR o ochrane osobných údajov vydané Európskym parlamentom [15, 9].

Klasifikačný proces je síce komplexný, avšak vytvorené politiky majú byť jednoduché. Štandard ISO/IEC 27001 slúži najmä k usmerneniu a každá organizácia má slobodu v tom ako si jednotlivé politiky zadefinuje a ako bude pracovať s klasifikovanými informáciami.

Prekážky klasifikovania a ochrany informácií

Nič nieje jednoduché a pri úsilí chrániť dáta nám vstupujú do cesty rôzne prekážky [1]. Medzi tie najčastejšie patria nasledujúce (obrázok 2.3):



Obr. 2.3: Najčastejšie prekážky klasifikovania a ochrany informácií

A. Ľudská chyba

Ako všetci veľmi dobre vieme, nikto z nás nie je dokonalý. Často sa nám stane, že niečo pokazíme buď z omylu, alebo z nevedomosti. Práve nedostatočné školenie zamestnancov ich privádza k otázkam typu:

1. Akou klasifikačnou triedou (úrovňou) mám klasifikovať dokument?
2. Kam mám vložiť klasifikačnú značku?

3. Aké klasifikačné triedy máme? Môžem si vymyslieť vlastnú?

4. Nemusím klasifikovať hneď teraz, alebo áno?

Toto všetko môže vyústiť do záveru, v ktorom sa informácie dostanú do nesprávnych rúk. Organizácie by preto mali mať vytvorený systém školení, ktorý by užívateľov naučil správnym postupom klasifikácie. Ak je to možné, je potrebné čo najviac uľahčiť klasifikáciu informácií. Dopredu vytvorené šablóny sú jedným zo skvelých spôsobov, ktorý naučí užívateľov klasifikovať tak ako sa to má. Klasifikácia a jej spôsob musí byť každému čo najviac jednoduchá a hlavne jednoznačná [15].

Je skvelé mať vybudované návyky aj pre klasifikáciu dokumentov. Klasifikovanie hneď pri ukladaní dokumentu je jednou z najlepších spôsobov, ako zabezpečiť dôveryhodnosť dokumentu po celú dobu jeho existencie. Nástroje pre kontrolu klasifikovania nám v tom napomáhajú. Môžu nám dať voľbu klasifikácie dokumentov pri ich ukladaní, či tlači. Podľa typu nastavenia nástroja sa môže napríklad aktivovať okno s voľbou klasifikačnej úrovne a my máme možnosť vybrať si úroveň dôvernosti dokumentu, ale takisto dostaneme aj možnosť odmietnuť okamžitú klasifikáciu. V režime povinnej klasifikácie nám túto voľbu znemožní a dokument sa stáva chráneným hneď [1, 15].

B. Politika organizácie

Pri vytváraní klasifikačných schém sa neraz dostávame k nezhode schém v jednotlivých oddeleniach organizácie. Môže sa stať, že dve schémy budú mať rovnako pomenované úrovne ale každá z nich bude predstavovať niečo iné. Politika v organizácii musí byť nastavená tak, aby spoločnosť mala len jednu klasifikačnú schému. Inak by mohlo dôjsť k nezrovnalostiam pri výbere úrovne, ktorou sa budú informácie klasifikovať [14]. Takisto pri spolupráci s dodávateľmi, respektíve tretími stranami, ktoré pracujú s informáciami zdieľanými s organizáciou, je potrebné oboznámiť každého o aktuálnej politike spoločnosti. V prípade, že rozsah ISMS nedokáže pokryť celú organizáciu, je vhodné založiť skupinu ľudí, ktorí budú pracovať na spoznaní vzťahov medzi jednotlivými časťami organizácie a po ich pochopení budú schopní vytvoriť podmienky na spoluprácu medzi jednotlivými oddeleniami. Ak ISMS je v niektorých častiach spoločnosti rozličný, potom sú tieto časti vnímané rovnako ako vonkajší dodávatelia, či tretie strany [1, 10].

C. Zdroje

Informačná bezpečnosť v rámci organizácie nie je vôbec lacná záležitosť. Školenie zamestnancov je náročné nielen časovo, ale aj finančne. Aby boli informácie zabezpečené čo najlepšie, často sa spoločnosti uchyľujú k nákupu užitočných nástrojov. Sú nápomocné a niekedy dokážu spoločnosti zachrániť krk. Avšak, tak, ako to už v dnešnom svete býva, nič nie je zadarmo a to platí aj pre tieto nástroje. Občas sa stane, že je potrebné klasifikačné schémy prerobiť a znovu zadefinovať ISMS organizácie. Výdavky, ktoré sa vložia do zabezpečenia informácií sú ale vždy menšie ako náklady, ktoré vzniknú pri ich úniku [1].

D. Technológia

Žijeme v treťom tisícročí, v období, v ktorom sa snažíme prenechať všetko, čo sa len dá, na strojoch. Vyvíjame stále presnejšie a rýchlejšie technológie zjednodušovania nášho života. Je niekoľko nástrojov, ktoré sú schopné oklasifikovať aktíva a takisto obmedzovať prístup k nim, čím tvoria ich ochranu. Táto technológia je dostatočne presná na to, aby mohla byť

bežne používaná, avšak pri automatickej klasifikácii aktív je stále nutná manuálna kontrola. Regulárne výrazy, rôzne pravidlá, slovníky a ich kombinácie nedokážu pokryť všetky prípady obsahu aktív, ktoré dokáže človek vyprodukovať.

Aj keby boli nástroje perfektné, môže sa stať, že z akéhosi dôvodu spravia chybu. Môže dôjsť napríklad k viacnásobnej klasifikácii a systém pre kontrolu prístupu k takémuto aktívu následne nebude môcť rozhodnúť, ktorá z daných úrovni klasifikácií je platná. To isté platí o opaku, kedy v metadátach aktíva sa nemusí značka uložiť a informácie sa ihneď stanú verejnými a absolútne nechránenými [1].

Elektronicky ukladané informácie musia mať označovanie vykonané takým spôsobom aby to technológia dokázala rozoznať. V prípade údajov ukladaných v databázach je vylúčené klasifikovať každú bunku a zabezpečiť jej ochranu pri agregácii dát. Nad takýmito systémami je potrebné vybudovať vyššiu vrstvu (napríklad rozumný framework), ktorá by bola schopná kontrolovať prístup k jednotlivým informáciám. Táto prezenčná vrstva však musí byť samostatne chránená [13].

Je veľmi problematické vytvoriť ochranu dát, ktoré sa nachádzajú napríklad na virtuálnych strojoch, či prenosných zariadeniach, ktoré je možné dostať mimo bezpečné prostredie organizácie. V týchto prípadoch musí byť vybudovaná dostatočne dobrá politika spoločnosti. Je odporúčané využívať systémy RMS (Record Management Systems), ktoré dokážu detekovať vytváranie, zmeny a mazanie záznamov. V spojení s inými nástrojmi je následne možné pomocou vhodne vytvorených pravidiel zabrániť rôznym incidentom vedúcim k úniku informácií [13].

E. Klasifikácia historických dát

Len málokedy sa stane, že spoločnosť už od svojho začiatku klasifikuje dáta. Po vytvorení ISMS v organizácii je tak množstvo takzvaných historických dát, ktoré nie sú klasifikované. Je jednoduché klasifikovať nové dokumenty pri ich vytvorení, uložení zmien a tiež pri tlači dokumentu. Ak by však spoločnosť chcela manuálne oklasifikovať všetky dáta, ktoré vznikli ešte pred začatím klasifikácie, tak by to bolo veľmi náročné. Využívajú sa preto nástroje, ktoré sú schopné hromadne oklasifikovať kompletne celé priečinky dokumentov na užívateľom zadanú klasifikačnú úroveň. Samozrejme užívateľ nemusí poznať obsah všetkých dokumentov a tu vzniká otázka voľby klasifikačnej úrovne. Tento problém je možné vyriešiť definovaním určitých pravidiel, ktorými bude nástroj rozlišovať hodnotu obsahu jednotlivých dokumentov. Podľa tejto hodnoty následne môže určiť klasifikačnú triedu dokumentu a automaticky klasifikovať hromadne množstvo dokumentov [13].

F. Klasifikácia dát v cloude

Informácie nevznikajú iba na lokálnych úložiskách ale aj v cloude, či informačných systémoch. Ako však oklasifikovať takéto dokumenty? Vytváranie dokumentov v cloudových úložiskách je často riešené pomocou aplikácií ako sú Google Docs a MS Office 365. Pre odstránenie nášho problému je teda možné napojiť tieto aplikácie na klasifikačnú službu, ktorá zabezpečí klasifikovanie dokumentov a uloženie klasifikačnej značky v metadátach dokumentu [13].

V predchádzajúcej časti práce sme si priblížili teóriu klasifikácie. Teraz je nutné uviesť ju do praxe. Vytvorenie riešenia, ktoré dokáže naplniť potreby užívateľov, si však vyžaduje spraviť dôkladný návrh. Analýza nám pri tom môže poskytnúť množstvo použiteľných nápadov. Analýza a návrh tak spolu vytvárajú základ, bez ktorého by aj najlepšia stavba padla,

ak by nebol dobre postavený. Pozrieme sa teda na niekoľko nástrojov, ktoré už klasifikovať dokážu.

2.3 Prehľad existujúcich nástrojov

Na trhu existuje množstvo nástrojov poskytujúcich klasifikáciu aktív, ktoré dokážu spolupracovať so systémom DLP (Data Loss Prevention). Väčšina z nich je tvorená ako rozšírenia do balíčkov Microsoft Office. Tieto nástroje zväčša aj vkladajú označenie klasifikačnej značky na príslušné, užívateľom zvolené, miesto.

Boldon James

Nástroje spoločnosti Boldon James sú veľmi rozsiahle a ponúkajú širokú škálu možností klasifikácie.

Office Classifier umožňuje klasifikáciu dokumentov v prostredí MS Office. Klasifikačnú značku umožňuje vložiť ako do hlavičky a pätičky dokumentu, tak aj ako vodoznak. Spôsob vizuálneho zobrazenia a dodatočného popisu je možné ľubovoľne upraviť [6].

Email Classifier funguje podobne ako vyššie uvedený Office Classifier. Vkladá značku do hlavičky a pätičky. Pri vkladaní už klasifikovaného súboru do prílohy správy, tento nástroj automaticky detekuje úroveň klasifikácie a v prípade potreby je schopný navýšiť klasifikáciu správy. Toto platí aj pre súbory zabalené v archíve ZIP. Výhodou tohto nástroja je možnosť zabrániť chybnému odoslaniu správy osobe s nízkou dôvernosťou. Dialógové okno vás na možný omyl upozorní a tým pomôže zabrániť úniku dôverných informácií [3].

File Classifier sa stará o klasifikáciu dokumentov viditeľných v prieskumníkovi operačného systému Windows. Stačí kliknúť pravým na súbor, prípadne viac označených súborov, a zvoliť príslušnú úroveň klasifikácie. Pri otvorení súboru sa automaticky podľa aktuálneho nastavenia zobrazí príslušná vizuálna značka do dokumentu [4].

Mobile Classifier pracuje podobne ako email classifier ale na mobilných zariadeniach [5].

Classifier Reporting slúži na správu klasifikovania na úrovni lokálneho zariadenia alebo oddelenia organizácie. Tento nástroj zobrazuje štatistiku počtov a spôsobu klasifikácie emailov, dokumentov, súborov, politík tak, ako to užívateľ potrebuje. Zároveň zobrazuje štatistiku ignorovaných upozornení podľa nastavenej politiky a mnoho iných užitočných informácií pomáhajúcich zlepšiť aktuálne politiky klasifikovania informácií v organizácii. Samozrejme nechýba možnosť vytlačenia a exportovania prezeraných štatistík [2].

BolDon James ponúka aj niekoľko ďalších nástrojov pracujúcich na podobnom princípe ale pre rôzne iné aplikácie. Jeho nedostatkom je slabšia schopnosť automaticky detekovať potrebnú klasifikačnú úroveň.

Titus

Nástroj Titus je veľmi podobný nástrojom Boldon James. Dokáže vkladať značky do hlavičky, pätičky a nerobí mu problém ani detekcia odosielania emailov osobám s nízkymi privilégiami. Umožňuje jednoducho preniesť klasifikačnú politiku organizácie do spôsobu klasifikácie a kontroly jednotlivých dokumentov. V čom je ale odlišný, je silná funkcia automatickej klasifikácie. Pri odhalení výskytu určitého pravidla v dokumente automaticky ponúkne výber určitej klasifikačnej úrovne. Upozorňuje taktiež na uloženie dokumentu s nižšou klasifikačnou úrovňou ako by podľa jeho automatického ohodnotenia mal dokument mať.

Užívateľom ponúka rozsiahlu škálu dokumentov, ktoré umožňuje oklasifikovať aby pokryl čo najväčšie množstvo informácií. Poskytuje taktiež sledovanie priečinkov a hneď pri vytvorení alebo zmene dokumentu v priečinku tento dokument oklasifikuje príslušnou klasifikačnou úrovňou [21].

Digital Guardian

Tento nástroj patrí medzi špičku v svojej kategórii. Ponúka nielen klasifikáciu, či už manuálnu alebo automatickú, ale aj identifikáciu a sledovanie citlivých dát od ich vytvorenia, modifikácie a prenosu. Organizácia tak dokáže mať dokonalý prehľad o tom, ako je s ich dátami narábané, a to v zariadeniach, ale aj na cloudových úložiskách. Digital Guardian ponúka takisto služby pre dáta ukladané v databázach. Dokáže vytvoriť mapovanie citlivých dát na jednotlivé zariadenia. Organizácia má potom väčšiu možnosť zabezpečiť práve tie miesta, na ktorých sa takýchto informácií uchováva najviac.

Veľkou výhodou rozpoznávania klasifikačnej úrovne je spôsob, ktorým dáta dokáže ohodnocovať. Dáta totiž hodnotí nielen na základe obsahu ale dokáže rozpoznať aj kontext, v ktorom sa nachádzajú. Množstvo nástrojov má problémy s integráciou v iných operačných systémoch a najčastejšie tak fungujú iba na OS Windows. Digital Guardian je multiplatformový nástroj a je kompatibilný s Windowsom, Linuxom a rovnako dobre aj s Mac systémom. Klasifikáciu a ochranu dát prevádzkuje dokonca aj na sieti, čím sa stáva medzi najlepšie nástroje pre ochranu citlivých informácii [7].

Uvedené nástroje určite nie sú všetky, ktoré by sme mohli využiť. Majú svoje výhody ale určite sa nájdu aj nevýhody, ktoré znižujú možnosť ochrany dôverných informácií. Princíp automatickej klasifikácie je u všetkých nástrojov rovnaký. Vytvorenie pravidiel, ktorými sa hľadajú v dokumentoch citlivé dáta. Nasleduje prevencia dôverných dát pred zneužitím v podobe vizuálnej klasifikačnej značky a značky skrytej v metadátach. Spôsob vytvárania pravidiel v týchto nástrojoch sa mi nepodarilo nájsť.

2.4 Aplikácia DocTag a klasifikácia v MS Office

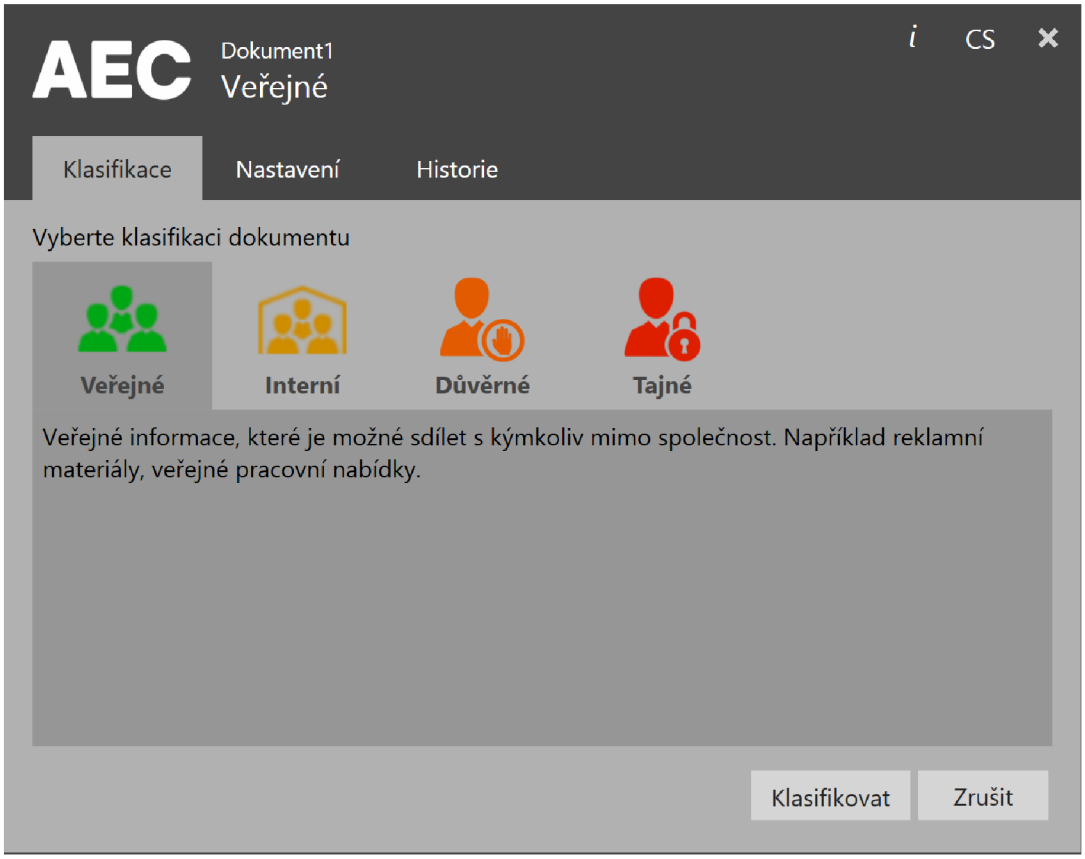
Spoločnosť AEC pred niekoľkými rokmi ukázala svetu svoje riešenie problému klasifikácie dokumentov. Je ním aplikácia DocTag, ktorá poskytuje užívateľom možnosť klasifikovať dokument. Jej používanie je v celku intuitívne a grafické prostredie je pre oko bežného človeka veľmi príjemné (obrázok 2.4). Podporuje dokumenty programov Word, PowerPoint, Excel, Outlook, Visio, Project a mimo prostredia MS Office aj rôzne ďalšie typy formátov. Táto aplikácia sa skladá z niekoľkých samostatných častí:

- webové rozhranie,
- konfiguračná služba,
- syslog služba,
- klientská aplikácia (Word, Excel, ...).

Vo webovom rozhraní sa vytvárajú a modifikujú konfigurácie pre jednotlivé skupiny zamestnancov podľa politiky danej organizácie. Prístup k tomuto rozhraniu má samozrejme iba istá skupina osôb spoločnosti. V konfiguráciách je možné nastavovať okrem iného aj klasifikačné triedy vrátane názvu, váhy, popisu, umiestnenia v dokumente a pod.

Konfiguračná služba predstavuje akési prepojenie webového rozhrania a klientskej aplikácie. Zachytáva zmeny v nastaveniach konfigurácií a konfiguráciu pre užívateľa konvertuje na XML zápis, aby klientská časť nemusela pracovať pri klasifikovaní s celou databázou ale iba s potrebnou konfiguráciou.

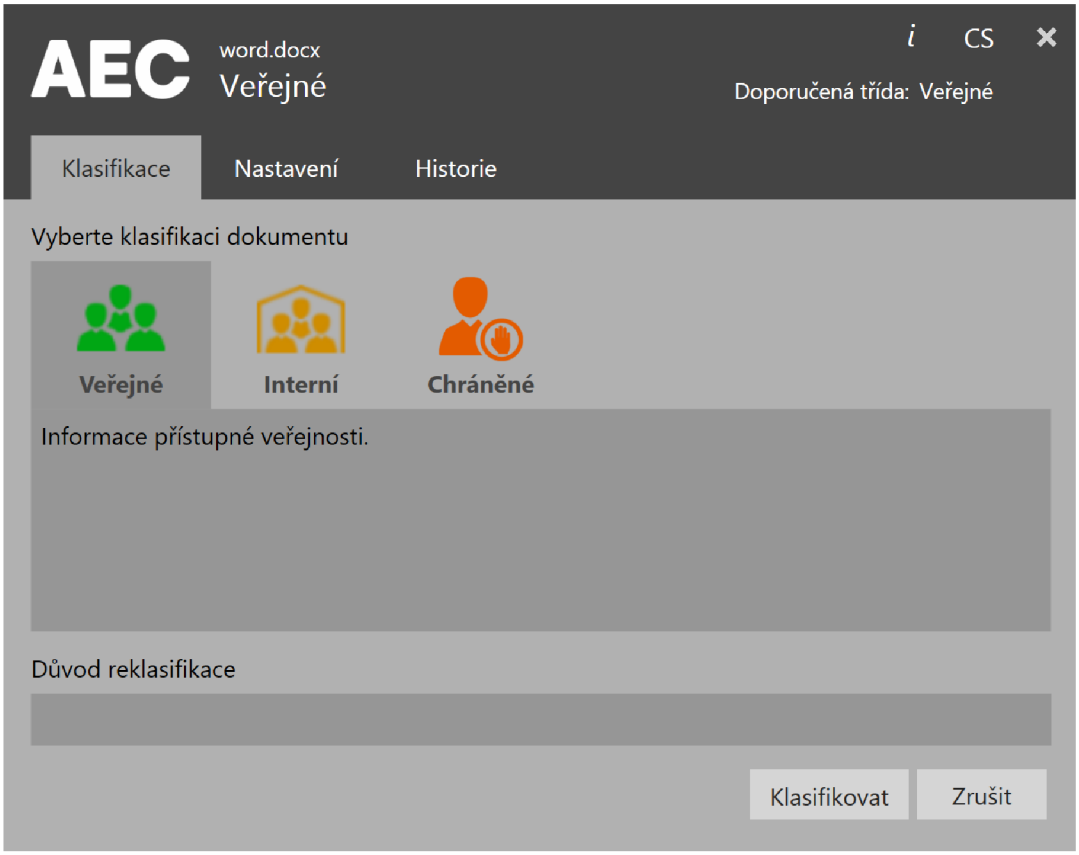
Klientská aplikácia je doplnok pre programy balíka Microsoft Office. Pri každom otvorení dokumentu z tohoto balíka aplikácia kontaktuje konfiguračnú službu, ktorá mu podá príslušnú konfiguráciu. Podľa režimu zvoleného v konfigurácii aplikácia pri ukladaní dokumentu môže vyzvať alebo dokonca prinútiť ku klasifikácii. Užívateľovi sa zobrazí okno pre klasifikáciu (viz. obrázok 2.4). Tu si zvolí triedu, ktorou chce klasifikovať dokument a príslušná značka sa následne umiestni viditeľne do hlavičky dokumentu a neviditeľne do metadát tohoto dokumentu. V rámci neviditeľnej značky je umiestnená takisto DLP značka, ktorá s použitím DLP systému bráni užívateľovi vykonávať isté úkony nad týmto dokumentom.



Obr. 2.4: Klasifikácia pomocou aplikácie DocTag

Služba Syslog je akýsi zapisovateľ udalostí, ktoré sa odohrali v klientskej aplikácii. Záznam o každej udalosti posiela v špeciálnom formáte do konfiguračnej databázy, ktorá zabezpečí, aby sa čo najrýchlejšie dostal tento záznam do databázy. Prehľad všetkých záznamov je možné spravovať vo webovom rozhraní.

Aby aplikácia DocTag zvýšila pravdepodobnosť správnej klasifikácie, ponúkne užívateľovi doporučenú klasifikačnú triedu. Doporučenie musí byť založené na základe analýzy obsahu dokumentu, ktorú je možné vykonať dvomi spôsobmi. Prvým je použiť umelú inteligenciu, ktorá sa postupným učením môže sama zdokonaliť až bude schopná určiť najvhodnejšiu klasifikačnú triedu. Druhým spôsobom je iterovať medzi sadou pravidiel pre jednotlivé triedy a spomedzi tých, v ktorých nájdeme zhodu, vybrať pravidlo, ktorému prislúcha klasifikačná trieda s najvyššou váhou. Názov tejto doporučenej triedy následne zobrazujeme textovou formovou v okne klasifikácie (obrázok 2.5). Analýza obsahu dokumentu v aplikácii DocTag je implementovaná v rámci tejto práce.



Obr. 2.5: Doporučenie klasifikačnej triedy v okne DocTag

Kapitola 3

Návrh a implementácia

Nástroje v sekcii 2.3 sú vytvorené na vysokej úrovni. Automatická klasifikácia nie je vždy správna a užívateľské rozhranie je nedostatočne prehľadné a intuitívne. Hlavnou úlohou sa preto stáva vytvorenie architektúry, ktorá by bola schopná rozpoznať dôvernosť informácií uložených v dokumente. K tomu by nástroj mal vlastniť sadu pravidiel pre detekciu citlivých dát. Pravidlá samozrejme nemôžu byť vymyslené a užívateľ by mal mať možnosť si ich vytvárať pomocou regulárnych výrazov, kľúčových slov a podobne. Aby sme mu jeho prácu čo najviac uľahčili, je nutné pripraviť rozhranie, ktoré bude intuitívne a hlavne efektívne pri používaní. Zadávanie pravidiel bude využívať jednoduchý jazyk, ktorý je takisto potrebné navrhnúť. Systém by mal dokázať spolupracovať s doplnkom pre MS Office, ktorý je schopný ukladať informácie o klasifikácii priamo do jadra dokumentu. Rovnako tak je dôležité aby klasifikácia prebiehala čo najrýchlejšie. Pokiaľ by sa malo klasifikovať niekoľko stoviek dokumentov, čo je v dnešnej dobe bežné v ktorejkoľvek väčšej organizácii, je nepredstaviteľné, aby užívateľ musel čakať hodiny do dokončenia.

3.1 Návrh jazyka pre tvorenie pravidiel

Vytváranie pravidiel by bolo možné viacerými spôsobmi. Jedným z nich by bolo vytvorenie formuláru na odklikávanie jednotlivých typov kľúčových slov, zadávanie počtu ich výskytu atď. Tento spôsob by bol pre užívateľa dosť komplikovaný, a preto použijeme spôsob popísania pravidiel pomocou jazyka.
Aby bol jazyk čo najjednoduchší použijeme syntax založenú na bezkontextovej gramatike. Cieľom celého jazyka je zapísať pravidlo do jednej hlavnej premennej, pričom bude možné použiť pravidlá v iných premenných, rôzne typy kľúčových slov a takisto operátory operujúce nad nimi.

Spracovanie regulárnych výrazov

V dokumentoch sa medzi informáciami často vyskytujú rôzne citlivé hodnoty, ako je napríklad číslo bankového účtu, IBAN, rôzne kombinácie slov a číselných hodnôt a podobne. Užívateľ si preto môže definovať vlastné regulárne výrazy, ktoré použije pri vyhľadávaní dôverných informácií. Umožníme mu ich zapísať pomocou funkcie *regex()*. Táto funkcia vyžaduje minimálne jeden parameter a tým je samotný regulárny výraz vo forme reťazca, prípadne užívateľ môže využiť už vstavané regulárne výrazy a zadať iba tzv. RegexID. Príklad použitia funkcie *regex()* môžeme vidieť na ukážke 3.1.

```
regex("^[a-z0-9-]+$",PHONESK)
```

Ukážka 3.1: Hľadanie regulárnych výrazov

Zoznam vstavaných regulárnych výrazov, RegexID:

- BIRTHNUM – dátum narodenia
- PHONECZ – české telefónne čísla
- PHONESK – slovenské telefónne čísla
- BANKACCOUNTNUM – čísla bankových účtov
- PIN – PIN kód
- IBAN – číslo účtu v tvare IBAN
- STREETADDRESS – ulica a číslo popisné
- DATENUMERIC – dátum v numerickom zápise
- SWIFT – SWIFT kód

Kľúčové slová

Aby užívateľ nemusel pre každé kľúčové slovo vytvárať regulárny výraz, môže jednoducho použiť funkciu *keyword()*, prípadne funkciu *keywordSensitive()*. Ak chce užívateľ odhaliť konkrétnu adresu alebo slová, ako napríklad „zmluva“, „mzda“ alebo iné slová ukazujúce na dôležité informácie, táto funkcia je tá pravá. Ak užívateľ chce rozlišovať veľké a malé písmená, použije funkciu *keywordSensitive()*, v opačnom prípade funkciu *keyword()*. Opäť môže zadať ľubovoľný počet parametrov, pričom táto funkcia sa správa podobne ako slovníkové vyhľadávanie, čiže štandardne sa bude hľadať aspoň jedno zo zadaných slov (ukážka 3.2).

```
keyword("zmluva","mzda") | keywordSensitive("AAa","aaA")
```

Ukážka 3.2: Hľadanie kľúčových slov

Emailové adresy

Obdobou kľúčových slov je vyhľadávanie emailových adries. K tomu by mala slúžiť funkcia *email()*, ktorej úlohou je hľadanie reťazcov v tvare emailových adries. Funkcia by mohla byť použitá bez parametra alebo ako parameter by mohol byť reťazec udávajúci adresu domény (ukážka 3.3).

```
email() | email("gmail.com")
```

Ukážka 3.3: Hľadanie emailových adries

Slovníkové vyhľadávanie

Jazyk by mal disponovať slovníkmi obsahujúcimi hromadu kľúčových slov, ktoré môžu napomôcť k odhaleniu napríklad adresy alebo zoznamu mien.
Príklady funkcií pre slovníkové vyhľadávanie:

- **towncz()** - funkcia vyhľadáva výskyt reťazcov v tvare názvu českého mesta alebo obce.
- **namecz()** - funkcia hľadá zhodu v slovníku s českými krstnými menami
- **surnamecz()** - funkcia sa zameriava na najčastejšie české priezviská

Každá z týchto funkcií by vracala počet výskytov, ktoré odhalila v rámci dokumentu. Samozrejme, postupom času by bolo možné pridávať mnohé ďalšie vstavané slovníky. Avšak výhodnejším riešením je ponúknuť užívateľovi možnosť použiť vlastný slovník. K tomuto účelu slúži funkcia *dictionary()*, ktorá vyžaduje jeden parameter a to cestu k slovníku. Tento slovník by mal mať na každom riadku uložené kľúčové slovo. Príklad použitia môžeme vidieť na ukážke 3.4.

```
dictionary("PATH\TO\FILE")
```

Ukážka 3.4: Slovníkové vyhľadávanie kľúčových slov

Zátvorky

Pre prednostné vyhodnotenie skupiny pravidiel je potrebné používať zátvorky. V našom jazyku sa používajú okrúhle zátvorky „ (“ a „) “. V rámci operátorov (popíšeme si ich neskôr) majú prioritu 2.

Premenné

Pokiaľ by to užívateľ potreboval, môže si jednotlivé pravidlá ukladať do premenných. Tak, ako je vidieť na ukážke 3.5, zápis premennej začína vždy kľúčovým slovom „var“, nasleduje názov premennej, znak „=“ a následne pravidlo, ktoré chce užívateľ uložiť. Sekvencia sa ukončuje znakom „ ; “ . Zápis premenných je inšpirovaný jazykom JavaScript.

```
var nazovPremennej = <PRAVIDLO> ;
```

Ukážka 3.5: Ukladanie pravidiel do premenných

Operátory

Operátory sú nevyhnutnou súčasťou každého jazyka. Pre naplnenie všetkých užívateľských požiadaviek by náš jazyk mal obsahovať operátory z tabuľky 3.1.

Operátor	Priorita	Význam	Príklad
„ & “	4	konjunkcia	towncz() & regex(“^([0-9]-?){7,8}[0-9]$")
„ | “	4	disjunkcia	towncz() | regex(“^([0-9]-?){7,8}[0-9]$")
„ ! “	4	negácia	!towncz()
„ ^ “	4	vylučujúca disjunkcia	towncz() ^ regex(“^([0-9]-?){7,8}[0-9]$")
„ * “	3	viacnásobný výskyt	5 * towncz()
„ > “	3	väčšie ako	5 > towncz()
„ < “	3	menšie ako	5 < towncz()
„ >= “	3	väčšie alebo rovné ako	5 >= towncz()
„ <= “	3	menšie alebo rovné ako	5 <= towncz()
„ == “	3	rovné ako	5 == towncz()
„ = “	1	priradenie	var pravidlo = towncz();

Tabuľka 3.1: Prehľad operátorov jazyka pre tvorbu pravidiel

Komentáre

Ak by niekto mal záujem vložiť komentár, tak ho môže zadať pomocou znaku „ # “ mimo parameter funkcie. Všetko za týmto znakom až do konca riadka sa bude považovať za komentár (ukážka 3.6).

```
# KOMENTAR
```

Ukážka 3.6: Vkladanie komentárov do pravidiel

Vytvorenie pravidla

Konečné pravidlo vzniká až uložením do tzv. hlavnej premennej „RULE“. Pre lepšie pochopenie je nižšie uvedený príklad (ukážka 3.7) aj s použitím vyššie spomenutých funkcií a operátorov.

```
RULE = towncz() | regex("^([0-9]-?){7,8}[0-9]$") ;
```

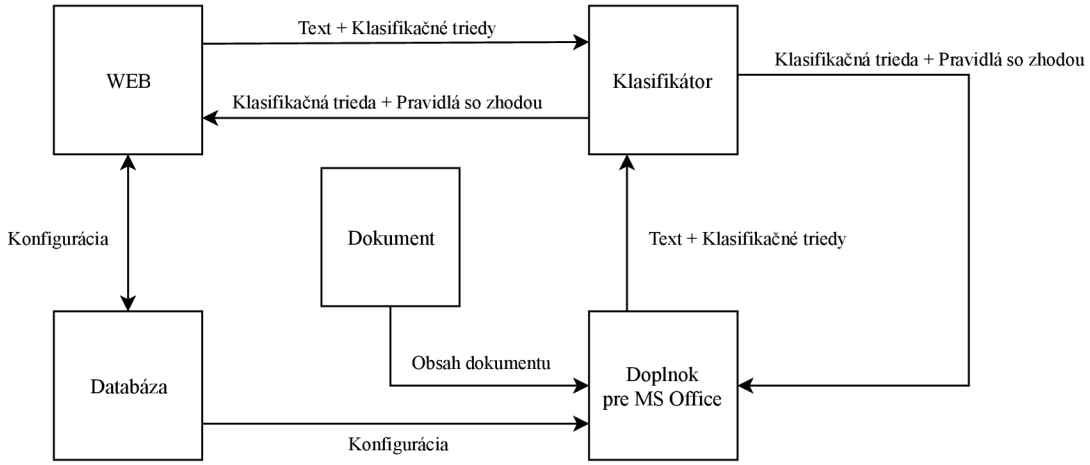
Ukážka 3.7: Ukladanie pravidla do hlavnej premennej RULE

Ukážka vytvárania pravidiel

Nižšie je uvedená triviálna ukážka (3.8) vytvorenia pravidla s použitím funkcií pre slovníkové vyhľadávanie, vyhľadávanie kľúčových slov, regulárnych výrazov a ich kombináciou pomocou operátorov.

```
var osoba = 5*namecz() | 3*email();
var hexValue = regex("^#?([a-f0-9]{6}|[a-f0-9]{3})$");
RULE = osoba & (hexValue | 42 * keyword("PRICE"));
```

Ukážka 3.8: Tvorba pravidiel



Obr. 3.1: Architektúra

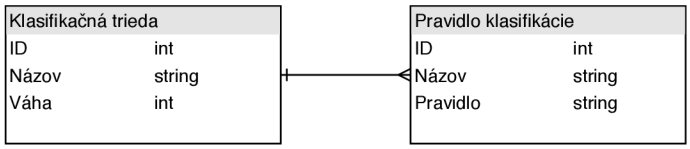
3.2 Návrh realizačnej schémy

Navrhnutá architektúra systému pre klasifikáciu je zobrazená na obrázku 3.1.

Vo webovom rozhraní sa nastaví konfigurácia klasifikovania. Určia sa klasifikačné triedy a aké pravidlá budú definovať ich výskyt. Táto konfigurácia sa uloží v databáze. V prostredí MS Office bude nainštalovaný doplnok, ktorý bude schopný klasifikovať tieto dokumenty. Doplnok pri otvorení dokumentu získa aktuálnu konfiguráciu, extrahuje text a ten pošle s jednotlivými triedami (každá obsahuje zoznam pravidiel) do klasifikátora. Danú triedu spolu so zoznamom pravidiel, ktorých zhodu našla v texte, predá naspäť doplnku v MS Office. Hneď, ako si užívateľ otvorí grafické rozhranie doplnku, bude mu pomocou textovej hlášky ponúknutá trieda. Následne môže doplnkom klasifikovať dokument touto alebo ním zvolenou inou triedou a zabezpečiť tak lepšiu ochranu informácií v tomto súbore.

3.3 Implementácia webového rozhrania

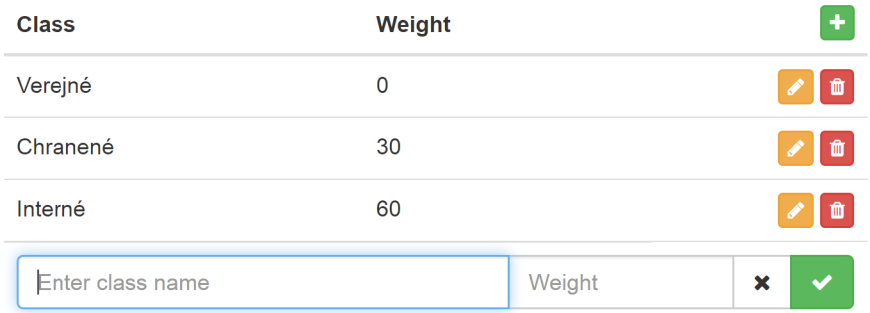
Jednotlivé organizácie majú odlišné politiky a dáta, ktoré nimi pretekajú sa môžu líšiť. Preto každá z nich si pre svoje záujmy bezpečia smie navrhnúť vlastnú konfiguráciu klasifikačných tried a pravidiel, pri ktorých splnení sa dáta označia touto triedou. Webové užívateľské rozhranie preto poskytuje možnosť vytvárania, modifikácie a mazania klasifikačných tried a aj príslušných pravidiel. Pri práci s formulármi je využitý *AJAX* a pre vytvorenie užívateľsky prívetivého prostredia som použil framework *Bootstrap*. Relačný model konfigurácie je zobrazený na obrázku 3.2.



Obr. 3.2: Relačný model konfigurácie

3.3.1 Klasifikačné triedy

Každá trieda obsahuje svoj názov, jednoznačné identifikačné číslo (slúži pre manipuláciu v databáze, užívateľ o ňom nevie) a váhu, ktorá určuje úroveň dôveryhodnosti dát označených danou triedou. Okrem týchto položiek triedy obsahujú aj zoznam pravidiel. Náhľad modifikácie klasifikačných tried je na obrázku 3.3.



Obr. 3.3: Modifikácia a vytváranie klasifikačných tried

3.3.2 Pravidlá

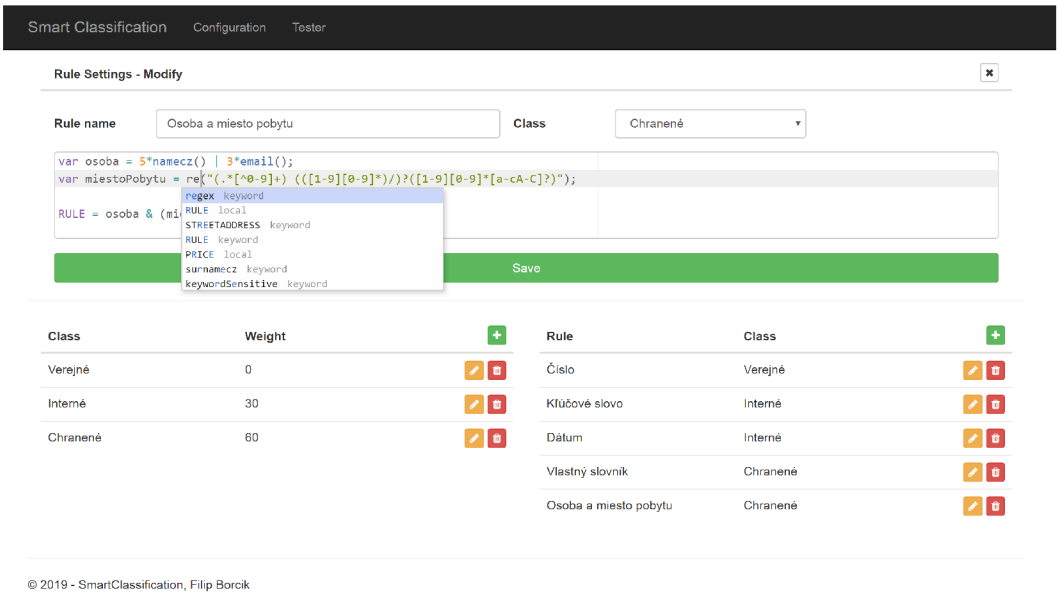
Klasifikátor sa musí riadiť istými pravidlami aby mohol určiť triedu dokumentu (textu, ktorý dokument obsahuje). Tie si nesmie vymyslieť ale musí ich navrhnúť osoba, ktorá dobre pozná politiku danej organizácie.

Pravidlá obsahujú svoje identifikačné číslo (platí to isté ako pre kl. triedu), názov, zápis samotného pravidla a sú priradené k práve jednej klasifikačnej triede.

Aby bolo písanie pravidiel jednoduchšie, k ich zápisu je použitý editor, ktorý zvýrazňuje text a ponúka kľúčové slová pre volanie funkcií či vkladanie názvov premenných (Obr. 3.4). K implementácii som využil zásuvný modul Ace editor [1], ktorý som rozšíril o syntax mnou navrhnutého jazyka.

Zápis syntaxe Ace editora sa podobá návrhu lexikálnej analýzy. Jedná sa o akýsi zoznam tokenov, kde každý obsahuje svoj názov, regulárny výraz, ktorým sa dá zapísať, a pole tokenov, ktoré môžu nasledovať za ním. Keďže Ace editor poskytuje validáciu syntaxe iba pre v ňom integrované jazyky, bolo nutné overovať správnosť zápisu pravidiel inou cestou. Pri chcení uložiť pravidlo sa zápis pošle do kontroléru na overenie a v prípade nesprávnej syntaxe klient príjme správu o chybe (Obr. 3.5). Postup overenia syntaxe v kontroléri si opíšeme neskôr.

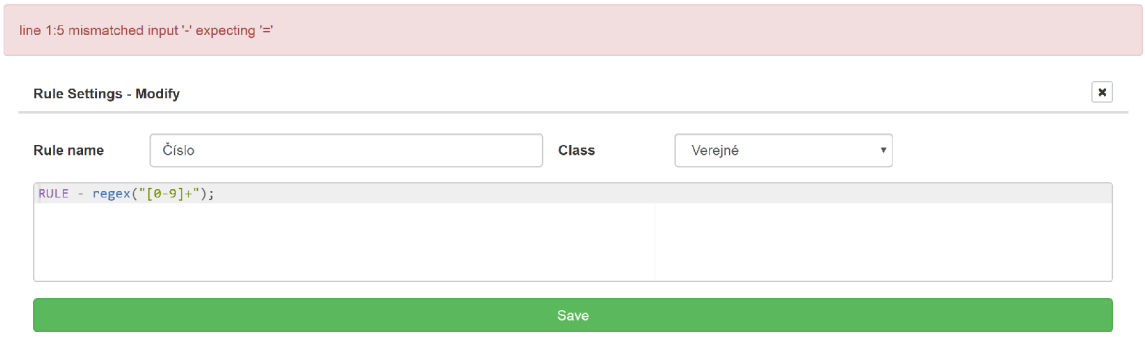
[1] Ace - The High Performance Code Editor For Web, `https://ace.c9.io/`



Obr. 3.4: Modifikácia a vytváranie pravidiel klasifikácie

3.3.3 Overovanie správnosti pravidiel užívateľom

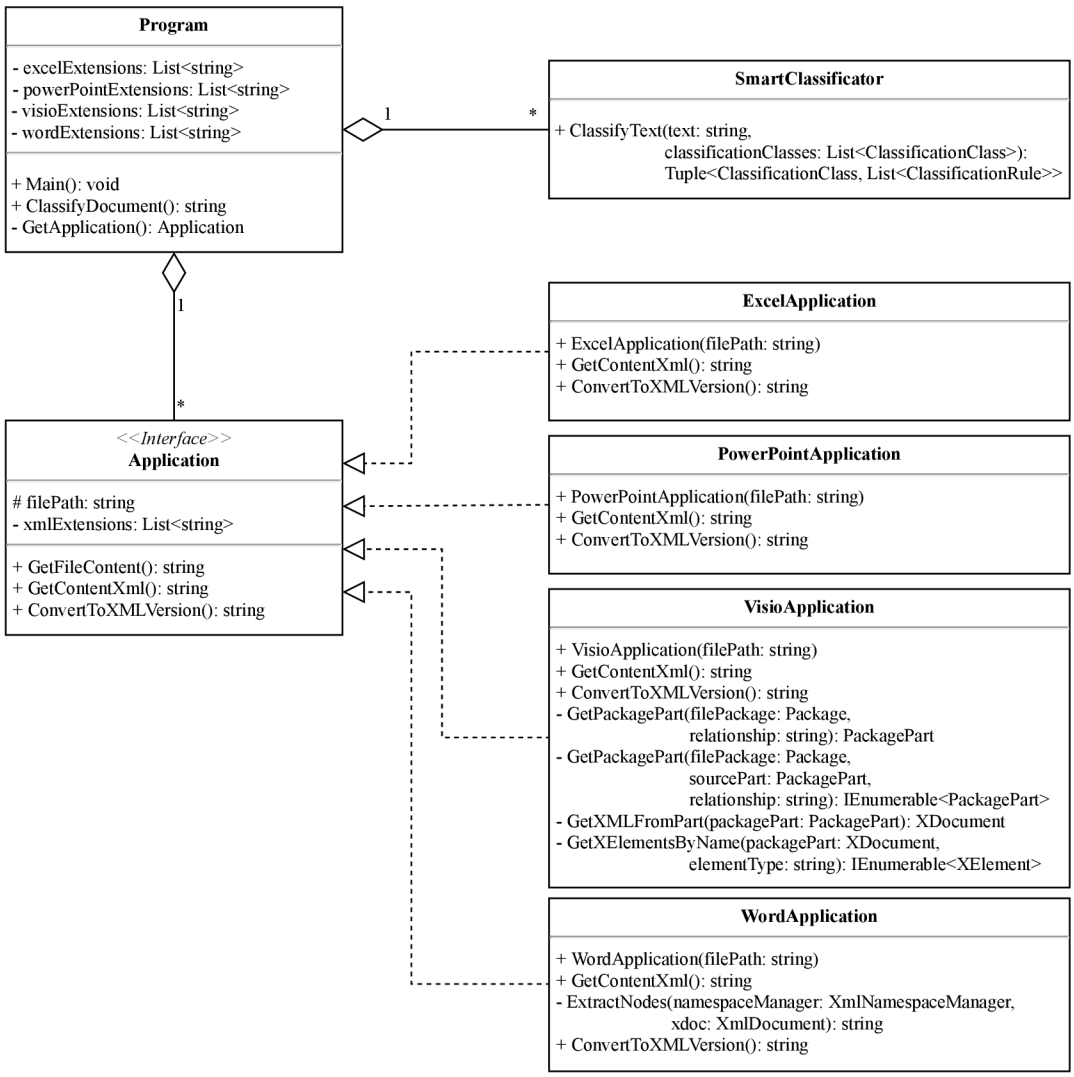
Často sa vraví, že sa najlepšie učíme na vlastných chybách. Avšak pokiaľ by sme chybne vytvorili pravidlá a následne kvôli tomu chybne označili dôverné dokumenty triedou verejné, tak by to pre firmu znamenalo únik dôležitých dát, čo sa kategorizuje ako bezpečnostný incident. Užívateľ má preto možnosť si pravidlá otestovať priamo vo webovom rozhraní. V okne tester smie vložiť akýkoľvek text do textového poľa a po spustení testu sa pošle tento text spolu s aktuálnou konfiguráciou do klasifikátora. Po klasifikovaní obsahu sa v príslušnom prvku zobrazí názov triedy, ktorá by bola ponúknutá aj v reálnom prostredí. Na pravej strane má samozrejme možnosť vidieť zoznam všetkých pravidiel, ktoré boli zhodné s obsahom textu. Pokiaľ by nebolo možné určiť triedu, v políčku sa zobrazí text *"None"*, v prípade chyby text *" Error"*. Ukážku výsledku testovania môžeme vidieť na obrázku A.1 v prílohe A.



Obr. 3.5: Správa o chybe v zápise pravidla

3.4 Rozpoznávanie klasifikačnej triedy dokumentov

Implementáciu komerčného produktu (doplnok pre MS Office) nebolo možné odovzdať, a preto, aby sme zachovali navrhnutú architektúru, musela byť do odovzdaného riešenia vložená cesta k dokumentu iným spôsobom. Simulovanie činnosti doplnku pre MS Office je v odovzdanom riešení implementované ako konzolová aplikácia.



Obr. 3.6: Diagram tried extrahovania a klasifikácie textu

3.4.1 Extrahovanie textového obsahu dokumentov

Aby sme vedeli, ktorý dokument chceme klasifikovať, definujeme cestu k nemu do premennej *path* v hlavnej metóde programu, *Main()*. Prvým krokom je volanie metódy *ClassifyDocument()* s cestou zadanou ako parameter. Spomenutá metóda na základe názvu súboru si vytvorí objekt príslušnej triedy pre spracovanie daného typu dokumentu. Tento krok je nutný z dôvodu rozdielneho prístupu jednotlivých aplikácii k práci s obsahom dokumentov. Bolo vytvorených 5 tried: WordApplication, ExcelApplication, VisioApplication, PowerPointApplication a Application, kde trieda Application je abstraktná a ostatné z nej dedia. Konštruktor týchto tried požaduje ako parameter cestu k súboru. Každá aplikácia balíčku MS Office má však na výber z dvoch hlavných spôsobov práce s dokumentmi. Táto implementácia vychádza z návrhového vzoru Strategy (Stratégia) a je popísaná diagramom tried na obrázku 3.6.

Prvým spôsobom je využitie knižnice *Microsoft.Office.Interop* [23]. Program sa môže chovať ako užívateľ, ktorý si spustí aplikáciu MS Office (Word, Excel, ...), v nej si otvorí chcený dokument a pomocou funkcií upravuje jeho obsah. Týchto funkcií je nespočetné množstvo a v každej aplikácii je pre dosiahnutie rovnakého cieľa nutné s nimi pracovať úplne iným spôsobom. Avšak pre programátora je to oproti druhému spôsobu značné uľahčenie práce. Veľkou výhodou je kompatibilita s každým typom dokumentu, starým či novým ale aj vytvoreným v akejkoľvek aplikácii balíku MS Office. Hlavnou nevýhodou používania tejto knižnice je rýchlosť. Čas spustenia aplikácie pre daný dokument, ale aj volanie jednotlivých funkcií, je trochu časovo náročné. Preto ak je to možné, tak je vhodné uchýliť sa k iným metódam a teda v odovzdanom riešení tento spôsob nieje použitý.

Druhým spôsobom je použitie *Open XML*. Tento vývojový balíček je postavený na API *System.IO.Packaging* [22]. Ako už názov napovedá, aplikovanie Open XML je postavené na práci s XML, v ktorom Microsoft od roku 2010 dokumenty ukladá. Avšak, podporovanými aplikáciami sú zatiaľ iba Word, Excel a PowerPoint. Práca s dokumentmi aplikácie Visio je možné pomocou API *System.IO.Packaging*, preto sú aj dokumenty tejto aplikácie zahrnuté medzi nami podporované.

Máme cestu k súboru a objekt príslušnej triedy pre spracovanie nášho dokumentu. Nasleduje kontrola podporovaných typov súborov pre danú aplikáciu. Podporované sú iba XML verzie dokumentov a teda každý z podporovaných súborov musí mať jednu z nasledujúcich prípon:

- MS Word: *'.docx', '.dotx'*
- MS Excel: *'.xlsx', '.xltx'*
- MS PowerPoint: *'.pptx', '.potx'*
- MS Visio: *'.vsdx', '.vstx'*

Pokiaľ súbor cez kontrolu neprejde vyvoláme výnimku hovoriacu o nepodporovanom súbore, avšak ak je všetko ako má prejdeme priamo k extrahovaniu obsahu pomocou Open XML.

Dokumenty MS Word

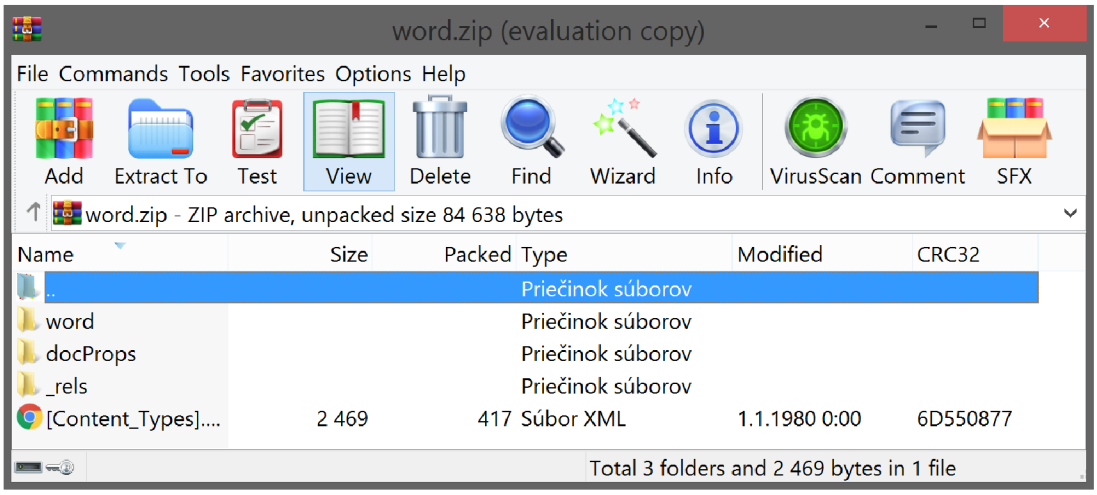
Spracovanie dokumentu aplikácie MS Word prebieha takto:

Vytvoríme si objekt triedy *StringBuilder*, ktorý nám zabezpečí správnu konkatenáciu nájdených reťazcov. Pomocou triedy *WordProcessingDocument* z balíčka Open XML si otvoríme súbor na čítanie. Každý XML dokument MS Office je v podstate iba ZIP archív zložený z niekoľkých priečinkov a množstva súborov, ktoré popisujú obsah, nastavenia, štýly a rôzne metadáta dokumentu (viz. obrázok 3.7, obrázok 3.8). Nás medzi týmito súbormi zaujímajú tie, ktoré obsahujú textový obsah dokumentu vrátane hlavičiek a pätičiek. Tieto sú uložené v zložke *"word"* a vieme sa k nim dostať pomocou vlastnosti *MainDocumentPart* nachádzajúcej sa v triede *WordProcessingDocument*. Vytvoríme si správcu pre XML menný priestor (ďalej iba namespace) a pridáme do neho URI namespace-u [2] s prefixom prvkov "w", keďže každý z XML prvkov, v nami hľadaných súboroch, je označený týmto prefixom, ktorý je popísaný spomenutým namespace-om. Do objektu triedy *XmlDocument* si načítame obsah pre nás zaujímavých súborov. Ďalej už iba hľadáme paragrafy, ktoré môžu obsahovať textové uzly. Tieto uzly sú označené názvom *"w:p"* a textové uzly sú označené názvom *"w:t"* [22] (obrázok 3.9). Obsah textových uzlov vkladáme do StringBuilderu na konkatenáciu. Postupne tak získame celý textový obsah dokumentu.

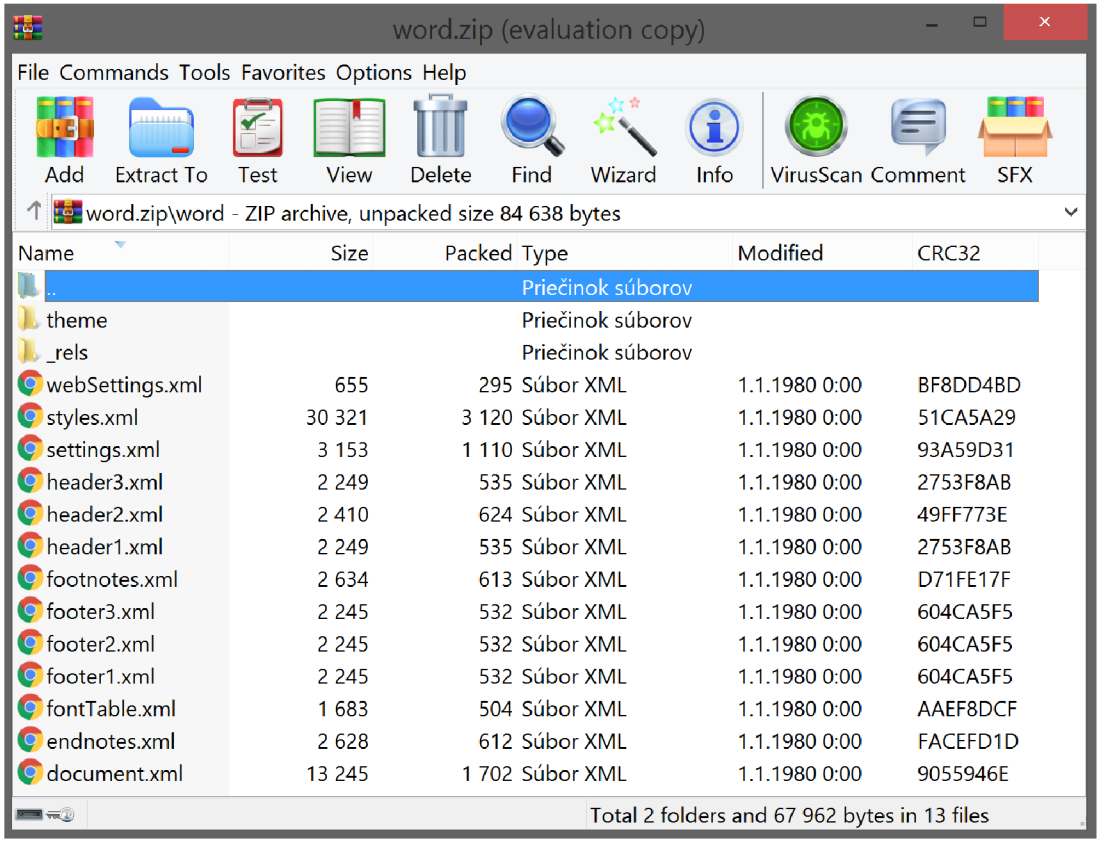
```
<w:p w:rsidR="00131432" w:rsidRDefault="00131432">
    <w:pPr>
        <w:pStyle w:val="Hlavicka"/>
    </w:pPr>
    <w:r>
        <w:t xml:space="preserve">Text vlozeny do </w:t>
    </w:r>
    <w:r>
        <w:t>hlavicky</w:t>
    </w:r>
    <w:bookmarkStart w:id="0" w:name="_GoBack"/>
    <w:bookmarkEnd w:id="0"/>
</w:p>
```

Ukážka 3.9: Časť dokumentu MS Word (súbor header2.xml) uložená v XML zápise

[2] namespace – http://schemas.openxmlformats.org/wordprocessingml/2006/main



Obr. 3.7: Štruktúra archívu dokumentu MS Word



Obr. 3.8: Štruktúra zložky obsahujúcej textový obsah dokumentu MS Word

Dokumenty MS Excel

Práca so súbormi MS Excel je podobná. Opäť prebieha otvorenie ZIP súboru na čítanie, avšak trieda sa mení na *SpreadsheetDocument*. V získanom objekte nás zaujíma vlastnosť *WorkbookPart* predstavujúca obsah súboru. Každý Excel dokument sa skladá z jednej alebo viacerých listov. List obsahuje 3 hlavičky a pätičky (vľavo, v strede a vpravo) a zároveň obsahuje bunky usporiadané do riadkov a stĺpcov. Zatiaľčo v hlavičkách a pätičkách je text uložený priamo vo vlastnostiach *InnerText* v jednotlivých bunkách tieto vlastnosti obsahujú iba čísla. Tieto čísla určujú pozíciu v tzv. zdieľanej tabuľke reťazcov (*SharedStringTable*), ktorú si drží vyššie spomenutá vlastnosť *WorkbookPart*. Postupne tak iterujeme medzi bunkami, hlavičkami a pätičkami až si vyskladáme úplný obsah dokumentu [22].

Dokumenty MS PowerPoint

Tieto dokumenty sa skladajú zo snímok, do ktorých môžeme vkladať text, obrázky, grafy, tabuľky a mnoho iného. Po otvorení ZIP súboru sa vytvorí objekt triedy *PresentationDocument*, z ktorého potrebujeme vlastnosť *PresentationPart*. Jednotlivé snímky tu ale nie sú prístupné priamo ale máme iba ich identifikačné čísla. Iterujeme teda listom týchto identifikátorov a obsah jednotlivých snímok si získavame z vlastnosti *PresentationPart* pomocou aktuálneho ID. Snímky majú text uložený podobne ako MS Word v textových uzloch, ktoré sú uložené v uzloch paragrafov. Tu už to však nemusíme robiť tak zložite a stačí použiť jednu z výhod jazyka C#, *LINQ*, a z aktuálnych snímok si získame potomkov, ktoré majú typ triedy *DocumentFormat.OpenXml.Drawing.Paragraph* a pri iterácii týmito paragrafmi si rovnakým spôsobom získame potomkov triedy *DocumentFormat.OpenXml.Drawing.Text*.

Dokumenty MS Visio

Tieto dokumenty sa skladajú zo strán, pričom obsah každej strany je tvorený takzvanými tvarmi. Tvary môžu obsahovať tabuľky, text, grafy alebo iné útvary. K extrahovaniu textu sme použili prístup doporučený spoločnosťou Microsoft [3]. Text, ktorý sme získali pomocou iterácie medzi jednotlivými tvarmi sme pre presnosť budúcej klasifikácie oddeľovali medzerou.

3.4.2 Analýza textu pravidlami

V predchádzajúcich častiach bol niekoľkokrát spomenutý tzv. klasifikátor. Jedná sa o triedu, ktorá má na starosti skontrolovať každé pravidlo v klasifikačných triedach, ktoré prijme medzi parametrami. Vykonáva sa kontrola validity pravidla a následne kontrola výskytu pravidla v texte (takisto zadaný v parametroch klasifikátora). Klasifikátor na výstupe posiela dvojicu *<ClassificationClass, List<ClassificationRule> >*. Klasifikátor je rovnaký pre webové rozhranie aj pre doplnok aplikácií MS Office.

Kontrola validity pravidiel

V užívateľskom rozhraní pre vytváranie a modifikáciu pravidiel sme použili doplnok *Ace editor* pre zvýrazňovanie syntaxe. Avšak jej kontrolu nezvláda a preto sa aj v module WEB používa pre kontrolu rovnaká metodika.

[3] `https://docs.microsoft.com/en-us/office/client-developer/visio/how-to-manipulate-the-visio-file-format-programmatically`

Keďže pravidlá pre klasifikáciu sú zapísané v rozšírenej BNF (Backus-Naurová forma), je možné využiť doplnok *Antlr*. Jeho verzia *Antlr4* [20] pre nás zohráva jednu z hlavných úloh. Pre jeho fungovanie je nutné vytvoriť súbor s našou gramatikou, ktorý obsahuje zoznam tokenov pre lexikálnu analýzu a zoznam pravidiel pre parser. Doplnok Antlr si zo zadanej gramatiky vytvorí dve triedy. Ako je možné očakávať, sú to triedy lexer a parser, ktorých objekty používame v implementácii. Pre vznik objektu lexer je nutné zadať ako parameter objekt triedy *AntlrInputStream*, vytvorenej z reťazca nášho pravidla pre klasifikáciu. Objekt parser vzniká s jediným parametrom, a to streamom tokenov z lexera.

Aby sme dosiahli lepšej prehľadnosti chýb v syntaxi pravidiel, bolo nutné si napísať triedu *CLSRulesErrorListener* dediacu z Antlr triedy pre kontrolu pravidiel a upraviť metódu pre výpis syntaktických chýb. Novovytvorený objekt typu tejto triedy nastavíme ako predvolený error listener pre parser. Následne pomocou metódy *parse()* získame usporiadaný strom tokenov.

Prechádzaním (zhora nadol) pomocou opäť nami implementovanej triedy *CLSRulesVisitor* dokážeme v správnom poradí vykonávať to, čo potrebujeme. Aby sme optimalizovali priebeh klasifikovania, ešte pred iteráciou klasifikačnými triedami, si vytvoríme špeciálny objekt *RulesChecker*. Pre každú funkciu nášho jazyka pre pravidlá klasifikácie, je v objektoch triedy RulesChecker metóda, ktorá počíta a vracia počet výskytov. Zároveň si uchováva výsledky hľadania v slovníkoch, pričom kľúč slovníka je vždy hľadaná položka. Ak sa najbližšie objaví rovnaká požiadavka, výsledok sa už znovu neanalyzuje ale sa vráti hodnota uložená v slovníku. Rovnako tak si uchovávame aj hodnoty uložené v premenných. Vytvorený RulesChecker zadávame ako parameter do konštruktora objektu triedy CLSRulesVisitor každého pravidla. Tento objekt pre každý uzol v strome určuje pravdivostnú hodnotu podstromu podľa typu daného uzla (typu operácie). Uzly na koncoch stromov predstavujú samotné funkcie a ich hodnota sa získava volaním metód objektu RulesChecker.

Postup pre testovanie validity zápisu pravidiel vo webovom rozhraní je založený na rovnakom princípe, avšak namiesto textu z dokumentov sa používa prázdny reťazec. Ak by sme nechceli kontrolovať aj sémantiku (hlavne existenciu premenných použitých vo výrazoch), stačilo by ak by sme si nechali vytvoriť strom tokenov a error listener by počas jeho vytvárania syntaktické chyby odhalil.

Proces hľadania kľúčových slov a regulárnych výrazov v texte

Vyhľadávanie kľúčových slov v texte vrátane slovníkového vyhľadávania, by bolo možné spracovávať viacerými spôsobmi. Našim požiadavkám najviac vyhovujúcim je použitie metódy jazyka C#, String.Split(). Niekomu by sa mohlo zdať divné hľadať kľúčové slová touto metódou, avšak mne sa zapáčila jej rýchlosť (viď test rýchlosti vyhľadávacích metód [17]) a aj efektivita zvládať problém zložených slov. Napríklad máme text “Annamária spadla.”. Hľadáme výskyt mien a náš slovník obsahuje mená Anna, Mária a aj Annamária. Pokiaľ by sme použili metódu Common Counting, String.Contains(), String.IndexOf(), Linq.Contains(), Linq.IndexOf() alebo hľadanie pomocou regulárnych výrazov (Regex je naviac aj dosť pomalý), tak by sme namiesto jedného výskytu objavili až 3 výskyty mien. Práve hľadanie pomocou metódy String.Split() je schopné zabezpečiť inkrementáciu počtu výsledkov iba pri nájdení prvého kľúčového slova v tomto zloženom slove. Rýchlosť však nie je ani s použitím tejto metódy dostatočná. Tento nedostatok je viditeľný najmä pri slovníkovom vyhľadávaní. Napr. priemerný čas vyhľadávania kľúčových slov zo slovníka obsahujúceho 1588 českých mien je 6 sekúnd.

Hľadanie rozšírených regulárnych výrazov je vykonávané metódou *Regex.Matches()*. Táto technika je dosť pomalá, preto sme efektivitu zvýšili pomocou kompilácie patternov na jednoduchšie. Zároveň sme obmedzili čas hľadania regulárnych výrazov na 750 ms. Tento časový interval bol určený na základe predpokladu, že ak máme 4 patterny schopné dosiahnuť tento interval, tak doporučenie klasifikačnej triedy dokumentu potrvá minimálne 3 sekundy. Žiadny užívateľ nerád čaká a bol by nespokojný ak by mal pri otvorení okna pre klasifikáciu dlho čakať. V prípade, že tento prípad nastane a vyhľadávanie regulárneho výrazu potrvá dlhšie ako 500 ms, tak metóda Regex.Match() hodí výnimku. Dlhá doba môže byť buď spôsobená veľmi veľkou dĺžkou textu alebo zle zapísaným regulárnym výrazom. Preto v tejto situácii teda rozdelíme reťazec na menšie 200 znakové časti, ktoré prechádzame paralelne vo viacerých vláknach a tentokrát s intervalom 100 ms. Ak sa výnimka opäť objaví, zrušíme beh týchto vlákien a skončíme s rozpoznávaním klasifikačnej triedy dokumentu. Rovnako postupujeme aj v prípade akejkoľvek inej chyby, ktorá nastane počas analýzy textu.

Kapitola 4

Testovanie a rozšírenia

Pre čo najlepšie vlastnosti každého produktu, je nutné tento produkt otestovať. V našom produkte je hlavnou požiadavkou rýchlosť, preto sme sa zameriavali hlavne na výkonnostné testy.

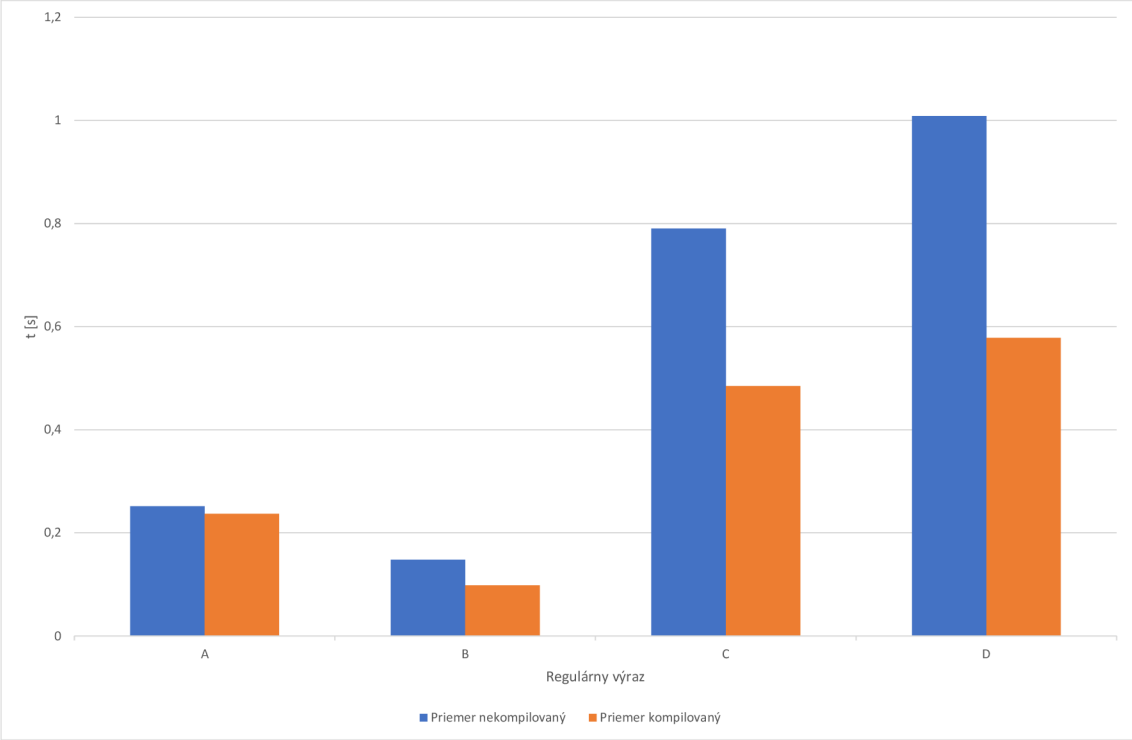
4.1 Výber správnej metódy hľadania regulárnych výrazov

Počas testovania funkčnosti klasifikátoru som narazil na problém. Vytvoril som si testovací dokument MS Word, lorem.docx, a jeho obsah som naplnil 302 stranami textu lorem ipsum. Na internete som našiel regulárny výraz pre adresu:
`"(.*[\wedge0-9]+) (([1-9][0-9]*)/)?([1-9][0-9]*[a-cA-C]?)"`. Následne som ho vložil ako pravidlo do klasifikátoru spolu s extrahovaným textom spomenutého súboru. Očakával som, že po maximálne niekoľkých sekundách budem mať výsledok, no nestalo sa tak ani po troch hodinách čakania, kedy som beh testu prerušil manuálne. Vtedy som si uvedomil, že nie je nijako ošetrený prípad, ak užívateľ do pravidla zadá podobný regulárny výraz. Prvá myšlienka ošetrenia bola prerušenie hľadania po istom časovom intervale a hneď druhou zrýchlenie procesu hľadania regexu.

4.1.1 Kompilovaný vs. nekompilovaný regulárny výraz

Trochu pátrania a možné riešenie bolo na svete. Takmer každý regulárny výraz je možné zjednodušiť prepísaním na niekoľko menších regulárnych výrazov [8]. Hľadanie jednoduchých regulárnych výrazov je podstatne rýchlejšie, a teda ak sa regulárny výraz správne upraví, je možné zrýchliť hľadanie u zložitejších aj o tretinu času. Avšak, kompilácia tiež neprebieha lusknutím prsta. Pokiaľ teda kompilujeme úplne jednoduchý regex, doba vykonávania môže narásť. Počítame ale s tým, že užívateľ zadá nielen jednoduché, ale aj zložité regulárne výrazy.

Priebeh testovania bol nasledovný. Vybrali sme si 4 regulárne výrazy, ktoré sme vyhľadávali v už spomenutom 302 stranovom dokumente lorem.docx s obsahom lorem ipsum. Každý regulárny výraz sme hľadali 100 krát bez kompilácie a rovnako tak 100 krát s kompiláciou. Priemery výsledkov sme zakreslili do grafu (Obr. 4.1).



Obr. 4.1: Kompilované vs. nekompilované regulárne výrazy

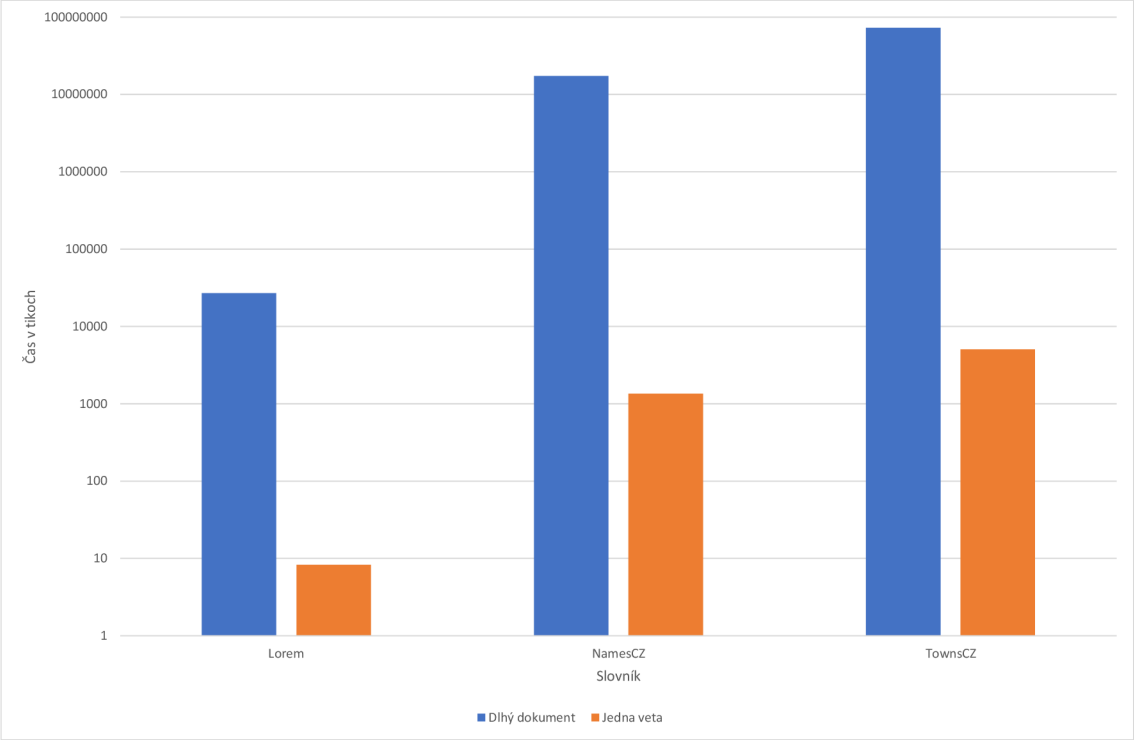
Zoznam regulárnych výrazov:

- A - `[a-z0-9_-]{3,16}`
- B - `#?([a-f0-9]{6}|[a-f0-9]{3})`
- C - `([a-z0-9_\.-]+)@([\da-z\.-]+)\.([a-z\.]{2,6})`
- D - `(https?:\/\/)?([\da-z\.-]+)\.([a-z\.]{2,6})([\/\w \.-]*)*\/?`

Na základe nášho testovania môžeme povedať, že celkový čas sa kompiláciou určite ušetrí, a to približne o 35 %.

4.2 Vplyv veľkosti slovníkov na rýchlosť vyhľadávania kľúčových slov

Pre presnosť analýzy klasifikačnej triedy dokumentov je vhodné mať čo najpodrobnejšie slovníky kľúčových slov, ktoré budú pokrývať danú triedu tak dobre, ako politika organizácie vyžaduje. Niekomu bude stačiť malý slovník, no niekto ho bude potrebovať mať extrémne podrobný. Implementovaný klasifikátor poskytuje tri vstavané slovníky, slovník českých mien, slovník českých obcí a slovník českých priezvisk. V rámci testovania rýchlosti využijeme slovník českých krstných mien a slovník českých obcí. Počet kľúčových slov, ktoré obsahujú, sa líši približne o štvornásobok. Zatiaľ čo prvý obsahuje 1588 položiek, ten druhý ich obsahuje 6256. Aby testovanie bolo ešte zaujímavejšie, do porovnania sme pridali aj vzorku obsahujúcu jedno kľúčové slovo, "lorem". Podobne ako v predchádzajúcom testovaní aj tu sme použili dokument lorem.docx a pre porovnanie aj druhý krátky dokument MS Word obsahujúci iba vetu: "Lorem ipsum dolor sit amet, consectetuer adipiscing elit.". Každú vzorku sme vyhľadávali 100 krát v oboch dokumentoch. Pretože niektoré výsledky dosahovali veľmi nízke hodnoty, čas sme nemerali v ms ale tikoch. Výsledky sú zobrazené v grafe na obrázku 4.2.



Obr. 4.2: Testovanie rýchlosti slovníkového vyhľadávania

Z výsledkov môžeme povedať, že procesorový čas strávený hľadaním kľúčových slov v dlhých dokumentoch je príliš veľký a je výhodnejšie rozdeliť dokument na viaceré menšie, ak je to možné.

4.3 Testovanie validity klasifikácie dokumentov

Pri tomto testovaní bolo vytvorených niekoľko dokumentov, v ktorých sme na rôzne miesta umiestnili istý text. Pozície neboli iba v hlavných častiach dokumentov, ale aj v hlavičkách, pätičkách a komentároch dokumentov. K jednotlivým súborom sme podľa textu priradili manuálne klasifikačné triedy, ktoré sme porovnávali s klasifikačnými triedami, ktoré rozpoznal náš klasifikátor. Analýza bola úspešná v prípade, keď klasifikátor klasifikoval dokument triedou s rovnakou alebo väčšou váhou a neúspešná pri opaku. Celkovo sme klasifikovali 20 dokumentov, z ktorých bolo 14 výsledkov zhodných s predpokladaným výsledkom, 6 dokumentov bolo klasifikovaných triedou s vyššou váhou ako bolo predpokladané a 0 dokumentov bolo klasifikovaných neúspešne (obrázok 4.3). Nástroj na klasifikáciu je teda pri dlhších textoch pomalý, ale v prípade správne zadaných pravidiel je dostatočne presný.

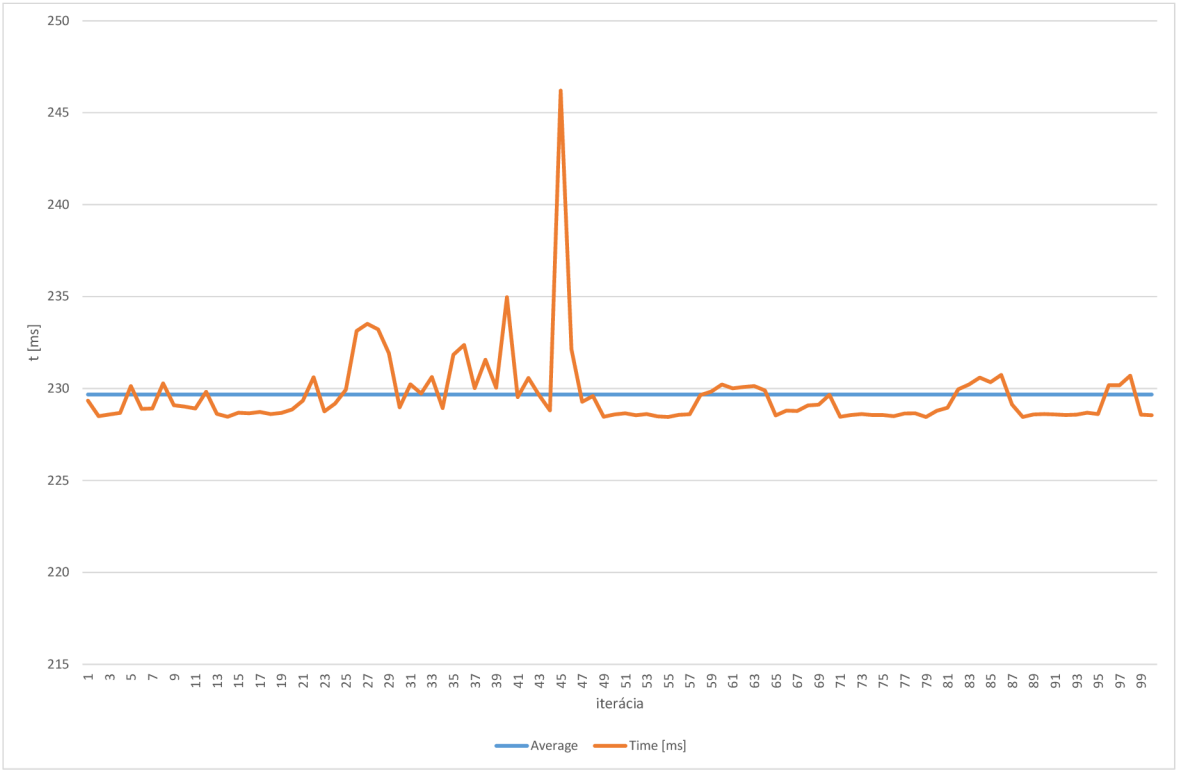
i	status	doc	expClass	resClass
1	OKAY	1.docx	Verejné	Interné
2	OKAY	2.docx	Chránené	Chránené
3	OKAY	3.docx	Chránené	Chránené
4	OKAY	4.docx	Chránené	Chránené
5	OKAY	5.docx	Chránené	Chránené
6	OKAY	6.docx	Chránené	Chránené
7	OKAY	7.docx	Verejné	Interné
8	OKAY	8.docx	Interné	Interné
9	OKAY	1.xlsx	Interné	Interné
10	OKAY	2.xlsx	Interné	Chránené
11	OKAY	3.xlsx	Interné	Chránené
12	OKAY	4.xlsx	Chránené	Chránené
13	OKAY	5.xlsx	Chránené	Chránené
14	OKAY	6.xlsx	Interné	Interné
15	OKAY	7.xlsx	Interné	Chránené
16	OKAY	1.pptx	Chránené	Chránené
17	OKAY	2.pptx	Chránené	Chránené
18	OKAY	3.pptx	Chránené	Chránené
19	OKAY	4.pptx	Chránené	Chránené
20	OKAY	5.pptx	Interné	Chránené

Obr. 4.3: Testovanie validity klasifikácie dokumentov

- i - poradie dokumentu
- status - výsledok testu
- doc - názov dokumentu
- expClass - očakávaná klasifikačná trieda
- resClass - výsledok klasifikácie

4.4 Testovanie rýchlosti klasifikovania dokumentov

Aby sme mali približný prehľad o rýchlosti analýzy, vytvorili sme si sadu dokumentov o veľkosti 1000. Klasifikovalo sa tromi triedami a celkovo 11 pravidlami. Keďže sme extrahovali text pomocou OpenXML nemusíme počítať s časom stráveným pri otváraní dokumentu v prostredí MS Office, a teda testovali sme iba rýchlosť čistej klasifikácie. Samozrejme táto rýchlosť sa mení v závislosti od počtu a zložitosti pravidiel a taktiež od veľkosti klasifikovaných dokumentov. Testovací set bol tvorený prevažne dokumentami s kratšou dĺžkou obsahu. V priemere bola rýchlosť klasifikácie celej sady 229675 ms a rýchlosť klasifikácie jedného dokumentu v sade 230 ms. Graf (Obr. 4.4) ukazuje priemerné časové trvanie klasifikácie jedného dokumentu v jednotlivých iteráciách.



Obr. 4.4: Testovanie rýchlosti klasifikovania dokumentov

4.5 Možné rozšírenia

Klasifikovanie týmto nástrojom je možné rozšíriť o niekoľko ďalších možností. Niektoré sú vhodné pre zvýšenie kompatibility, iné zase na presnosti klasifikácie.

4.5.1 Klasifikácia všetkých typov dokumentov MS Office

Ako už bolo vyššie uvedené, náš nástroj klasifikuje dokumenty aplikácií Word, Excel, PowerPoint a Visio. Keďže doplnok DocTag od otvorenia dokumentu pracuje s inštanciou aplikácie pomocou Office.Interop, mohli by sme to využiť. Pri dokumentoch typov, v ktorých

je možné s OpenXml pracovať (Word, Excel, PowerPoint, Visio), by sme mohli ich staršie verzie najskôr konvertovať na XML verziu uloženú v dočasných súboroch. Túto kópiu by sme analyzovali pomocou OpenXml a následne zmazali.

V dokumentoch iných typov aplikácií, by sme mohli text priamo extrahovať s Office.Interop, čím by sme pokryli celú škálu dokumentov Microsoft Office.

4.5.2 Overovanie odtlačkov

Ide o metódu, v ktorej by sme už z predchádzajúcich poznatkov o klasifikovaných dokumentoch mohli rýchlejšie rozpoznávať triedu iných dokumentov. Klasifikovaný dokument by sme rozdelili na časti, ktoré zahešovali a k tomuto hešu pridelili triedu, ktorou bol daný dokument klasifikovaný. Pokiaľ by sme našli novom dokument obsah, ktorý je po zahešovaní rovnaký, mohli by sme predpokladať, že jeho obsah je rovnako dôveryhodný, a teda mal by byť klasifikovaný rovnakou triedou. Táto metodika by bola vhodnejšia pri klasifikovaní pomocou umelej inteligencie, než pri klasifikovaní pomocou pravidiel.

4.5.3 Rozšírenie pravidiel

Naše pravidlá momentálne pokrývajú regulárne výrazy a kľúčové slová. Mohli by sme ich rozšíriť ale aj o iné prvky. Napríklad o rôzne vlastnosti dokumentu, medzi ktorými sú napríklad názov, veľkosť, umiestnenie a rôzne iné. Tieto parametre ale nemožno vyhodnocovať až v dobe iterácie medzi uzlami stromu pravidiel, pretože dokument už v danej dobe nie je známy a poznáme iba textový obsah dokumentu. Vlastnosti by sa museli teda vyhodnotiť podobne ako text už pred iteráciou stromom pravidiel a poslať sa do objektu *RulesChecker* ako ďalší parameter.

Kapitola 5

Záver

V tejto práci bola opísaná klasifikácia dokumentov tak, ako ju definuje rodina štandardov ISO/IEC 27000. V rámci práce bol implementovaný systém pre automatickú klasifikáciu dokumentov vytvorených aplikáciami *Word*, *Excel*, *PowerPoint* a *Visio*, ktoré sú súčasťou balíka *Microsoft Office*. Na základe analýzy spôsobov extrakcie dát z týchto dokumentov bolo zvolené použitie knižnice *OpenXML*, ktoré síce limitovalo klasifikáciu iba na novšie typy dokumentov ale umožnilo rýchlu extrakciu dát. Bolo vytvorené webové rozhranie pre modifikáciu konfigurácie klasifikačných tried a pravidiel, ktorými sú definované. Konfigurácia bola ukladaná do databáze čím bola prepojená s doplnkom DocTag od spoločnosti AEC. Doplnok po extrakcii textového obsahu dokumentu má za úlohu posielať text na analýzu klasifikátoru, ktorý na základe pravidiel uložených v konfigurácii určil najvhodnejšiu triedu klasifikácie dokumentu.

Testovaním sme zistili, že ak sú dobre definované pravidlá, klasifikácia týmto nástrojom je dostatočne presná. Avšak, vysoká presnosť si vybrala svoju daň a rozsiahle pravidlá v spojení s dlhším textom uloženým v dokumente značne spomaľujú čas klasifikácie. Toto je výrazná nevýhoda, ktorá obmedzuje plné použitie klasifikátoru v doplnku *DocTag* (je možné rýchlo klasifikovať iba menšie dokumenty). Pokiaľ by sa v budúcnosti nenašiel spôsob zrýchlenia, tak by klasifikátor mohol nájsť uplatnenie pri vytváraní dátových setov pre učenie umelej inteligencie slúžiacej na klasifikovanie dokumentov.

V rámci budúceho zlepšovania systému klasifikácie by bolo vhodné rozšíriť podporu typov klasifikovaných dokumentov. Medzi ďalšie rozšírenia patrí taktiež rozšírenie pravidiel o veľkosť, názov a umiestnenie dokumentu. Zároveň by mohla byť implementovaná analýza odtlačkov obsahu dokumentov, ktorá by mohla urýchliť proces klasifikácie.

Literatúra

[1] Bernsmed, K.; Fischer-Hübner, S.: *Secure IT Systems: 19th Nordic Conference, NordSec 2014, Tromsø, Norway, October 15-17, 2014, Proceedings*, ročník 8788. Springer International Publishing, 2014, ISBN 978-3-319-11598-6.

[2] Boldon James - Classifier Reporting. 2015.
URL `https://www.youtube.com/watch?v=QoOhBwx6qZU`

[3] Boldon James - Email Classifier. 2015.
URL `https://www.youtube.com/watch?v=oz2Iw5KBgN8`

[4] Boldon James - File Classifier. 2015.
URL `https://www.youtube.com/watch?v=mdGZx4TwmgI`

[5] Boldon James - Mobile Classifier. 2015.
URL `https://www.youtube.com/watch?v=lMGMVLzoz5U`

[6] Boldon James - Office Classifier. 2015.
URL `https://www.youtube.com/watch?v=Z4QnMyaOfao`

[7] Digital Guardian: What is Data Classification? A Data Classification Definition. 2018.
URL `https://digitalguardian.com/resources/data-security-knowledge-base/data-classification`

[8] Goyvaerts, J.: *Regular Expressions: The Complete Tutorial*. Lulu Press, 2006, ISBN 9781411677609.

[9] Information Classification and ISO 27001. December 2016.
URL
`https://www.doxonomy.com/blog/information-classification-and-iso-27001`

[10] Information Classification – Common Questions - CertiKit. August 2017.
URL `https://certikit.com/information-classification-common-questions/`

[11] International Organization for Standardization: *ISO/IEC 27001: 2013: Information Technology–Security Techniques–Information Security Management Systems–Requirements*. Geneva, CH: International Organization for Standardization, Október 2013.

[12] International Organization for Standardization: *ISO/IEC 27000: 2018: Information technology–Security techniques–Information security management systems–Overview and vocabulary* . Geneva, CH: International Organization for Standardization, Február 2018.

[13] Kacic, M.: Klasifikace informací v korporátním prostředí. *DSM*, 2018: s. 12–17, doi:https://aec.cz/cz/ztisku/matej-kacic-klasifikace-informaci-v-korporatnim-prostredi-dsm-2018.pdf.

[14] Kosutic, D.: Problems with defining a small ISMS scope in ISO 27001. Jún 2010.
URL `https://advisera.com/27001academy/blog/2010/06/29/problems-with-defining-the-scope-in-iso-27001/`

[15] Kosutic, D.: Information classification according to ISO 27001. Máj 2014.
URL `https://advisera.com/27001academy/blog/2014/05/12/information-classification-according-to-iso-27001/`

[16] Kosutic, D.: What is ISO 27001: The basic logic of information security management. *URL https://advisera.com/27001academy/knowledgebase/the-basic-logic-of-iso-27001-how-does-information-security-work/*,
2016.

[17] Lozinski, D.: C# .Net: Fastest Way to check if a string occurs within a string. 2013.
URL `http://cc.davelozinski.com/c-sharp/fastest-way-to-check-if-a-string-occurs-within-a-string`

[18] Mgr. Jan Matula, P.: ISO 9000, 20000, 27000. 2016.
URL `https://is.muni.cz/el/1421/podzim2016/VIKMA07/um/4__ISO_9000__20000__27000.pdf`

[19] Pandini, W.: ISO 27002: Best Practices for Information Security Management. *URL https://ostec.blog/en/general/iso-27002-best-practices-ism*, December 2016.

[20] Parr, T.: *The Definitive ANTLR 4 Reference*. Pragmatic Bookshelf, 2013, ISBN 9781934356999.

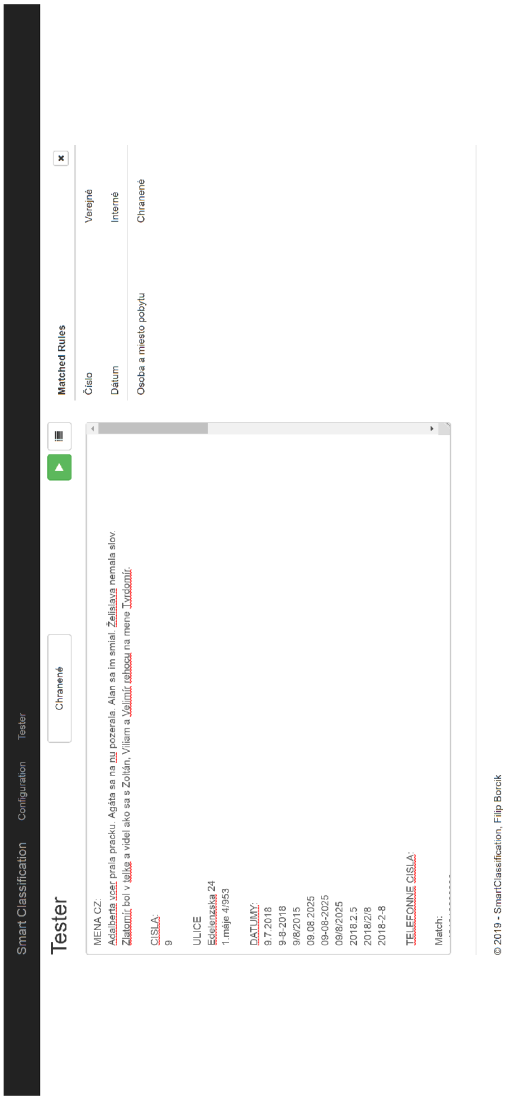
[21] TITUS: Data Classification.
URL `https://www.titus.com/data-classification`

[22] van Vugt, W.: *Open XML Explained*. Microsoft, 2007.

[23] Whitechapel, A.: *Microsoft® .NET Development for Microsoft Office*. Microsoft Press, 2005, ISBN 0735621322.

Príloha A

Obrázky, tabuľky a ukážky



Obr. A.1: Testovanie správnosti pravidiel klasifikácie

Zoznam obrázkov

Zoznam tabuliek

Zoznam ukážok

Príloha B

Obsah priloženého CD

Priložené CD obsahuje nasledujúce položky:

- text bakalárskej práce,
- zdrojové kódy pre LaTeX,
- zdrojové kódy programu.