*Witold Abramowicz, Elżbieta Bukowska, Jakub Dzikowski,*
*Agata Filipowska, Tomasz Kaczmarek, Jacek Małyszko, Bartosz Perkowski,*
*Krzysztof Węcel, Dawid Węckowski, Karol Wieloch, Piotr Stolarski*

Katedra Informatyki Ekonomicznej
Uniwersytet Ekonomiczny w Poznaniu

# ARCHITEKTURA SYSTEMU WYKRYWANIA ZAGROŻEŃ W CYBERPRZESTRZENI

## 1. Motywacja

Przedstawione w niniejszym artykule rozwiązanie architektoniczne zostało przyjęte w projekcie Semantyczny Monitoring Cyberprzestrzeni, realizowanym z grantu w Katedrze Informatyki Ekonomicznej na Uniwersytecie Ekonomicznym w Poznaniu. Celem projektu jest wykrywanie zagrożeń w Internecie, a w szczególności wykrywanie zagrożeń związanych z nielegalnym handlem lekami. Zagrożenia te manifestują się na wielu portalach ogłoszeniowych i forach internetowych, na których jest proponowana niezgodna z prawem sprzedaż leków. Bezpośrednim beneficjentem projektu ma być Komenda Główna Policji, w której właściwy departament mógłby powstający system wykorzystywać do monitorowania źródeł internetowych pod kątem tego typu zagrożeń. Możliwe są również zastosowania opracowanych w projekcie metod w innych branżach, np. związanych z monitorowaniem innego rodzaju zagrożeń dla Policji, monitorowania Internetu pod kątem wzmianek o produktach lub konkretnych firmach, co można wykorzystać w prowadzaniu kampanii PR lub w działalności marketingowej firm. Kluczowe wymagania wobec projektu wynikają z jego dziedziny i jej osadzenia w naukach sądowych i informatyki śledczej. W naukach sądowych jest obecne pojęcie procesu śledczego, określające sposób postępowania z materiałem dowodowym podczas prowadzenia śledztwa. Analogiczne pojęcie można zdefiniować na gruncie informatyki śledczej. Proces śledczy w informatyce

śledczej składa się z trzech głównych faz[1]: wyszukiwania i gromadzenia materiału dowodowego, jego klasyfikacji, łączenia i rekonstrukcji oraz prezentacji. Jednocześnie w ramach tych faz informatyka śledcza proponuje sposoby dokumentowania pracy z materiałem dowodowym oraz sposoby przechowywania dowodów elektronicznych. Architektura tworzonego systemu informatycznego odzwierciedla główne fazy procesu śledczego w informatyce śledczej.

## 2. Przegląd rozwiązań

Rozwiązania architektoniczne zastosowane w systemie są wzorowane na rozwiązaniach z dwóch obszarów: systemów wyszukiwawczych i systemów przetwarzania jezyka naturalnego. W obydwu tych grupach systemów stosuje się przetwarzanie potokowe – zarówno przy pobieraniu stron do indeksowania z sieci, jak i przy przetwarzaniu dokumentów przez kolejne algorytmy rozpoznawania struktur językowych. Szczegółowe rozwiązania w obu tych przypadkach omówiono poniżej.

### 2.1. Systemy wyszukiwawcze

W przypadku systemów wyszukiwawczych wzorowano się na ogólnym podejściu, w którym centralne znaczenie odgrywają dwa repozytoria – repozytorium stron oraz indeks. Oba te repozytoria są zasilane przez mechanizm pozyskania stron (*crawler*/robot) i poszczególne jego moduły stosowane sekwencyjnie. Istnieją dwie główne klasy robotów zbierających strony – przeznaczone do stron Internetu powierzchniowego oraz głębokiego. W przypadku Internetu powierzchniowego podstawowym rodzajem robota jest robot jednoprocesowy. Jest to najprostszy rodzaj robota, w którym poszczególne etapy przetwarzania są realizowane sekwencyjnie: rozwiązanie adresu URL na adres IP serwera, pobranie strony z adresu znalezionego w bazie adresów, parsowanie strony, wyodrębnienie kolejnych adresów do odwiedzenia i usunięcie duplikatów oraz wyodrębnienie tekstu. Bardziej zaawansowaną wersją jest robot równoległy, który wszystkie te procesy wykonuje dla różnych stron równocześnie, korzystając z jednej bazy adresów i jednego repozytorium pozyskanych treści oraz mechanizmu zarządzania kolejkami przetwarzania adresów w celu równoważenia obciążenia oraz spełnienia polityk poprawności wobec odwiedzanych serwerów[2].

[1] J. Ashcroft, *Electronic Crime Scene Investigation – A Guide for First Responders. Tech. rep. U.S. Department of Justice*, 2001; J. Dzikowski, *Wykrywanie przestępczości z wykorzystaniem informacji ze źródeł internetowych*, praca magisterska – Uniwersytet Ekonomiczny w Poznaniu, 2010; M. Pollitt, *Applying traditional forensic taxonomy to digital forensics,* „IFIP Int. Conf. Digital Forensics" 2008, vol. 285, red. I. Ray, S. Shenoi, Springer; M. Solomon, N. Broom, D. Barrett, *Computer Forensics JumpStart,* Alameda, CA, USA: SYBEX Inc., 2004.

[2] Ch. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, Cambridge 2008.

W skrajnym przypadku rozproszenie może oznaczać rozproszenie geograficzne lub dedykowane do różnych segmentów sieci – mamy wówczas do czynienia w istocie z wieloma robotami działającymi jednocześnie dla różnych segmentów sieci. Głównym problemem przy robotach równoległych jest to, jak ustalić, którą stronę najpierw należy odwiedzić. W przypadku robota wieloprocesowego tworzy to narzut na komunikację pomiędzy poszczególnymi instancjami robota.

Specyficzną klasą robotów są tzw. roboty ukierunkowane (ang. *focused crawler*). Ten typ robota stosuje filtr i stara się nie pobierać stron nierelewantnych. Ważnymi elementami architektury robota są agenty zbierające, historia zbieranych stron (umożliwia uniknięcie duplikacji) oraz indeks[3].

Drugim rodzajem robotów są roboty przeznaczone do zbierania stron z głębokiego Internetu, a więc baz danych udostępnianych w Internecie. W wielu publikacjach zaproponowano różne podejścia do tworzenia tego typu systemów. Anuradha i Sharma proponują architekturę, której główne elementy to mechanizm rozpoznawania interfejsu do bazy, mechanizm zgłaszania zapytań, mechanizm ekstrakcji danych z pozyskanych stron, interfejs zadawania zapytań do zgromadzonych danych i odpowiadania na nie[4]. Ceri prezentuje wizję nowego rodzaju systemów wyszukiwawczych, które umożliwiają zadawanie złożonych zapytań do wielu źródeł ukrytego Internetu, integrują wyniki oraz zapewniają optymalizację wykonania zapytań. Całościowa architektura jest bardzo rozbudowana i obejmuje interfejsy obsługi użytkownika, interfejsy do źródeł, mechanizmy obsługi pamięci podręcznej dla wyników zapytań, mechanizmy analizy zapytań oraz ich optymalizacji[5]. W jeszcze innym podejściu proponowana architektura zawiera interfejs do zadawania zapytań, mechanizm wypełniania formularzy w źródłach i analizator formularzy, mechanizm zbierania stron, ekstrakcji informacji ze stron, mechanizm ekstrakcji linków i ich klasyfikacji (autorzy proponują *crawler*, który może ustalić priorytet strony na podstawie jej adresu, a nie treści)[6].

Wspólnymi elementami tego rodzaju robotów są: konieczność dostosowania ich do obsługi wybranych źródeł (w tym wymagających wypełnienia formularzy), ukierunkowanie (zastosowanie filtra) i mechanizmy ekstrakcji informacji ze stron, które są istotne również w proponowanym rozwiązaniu.

---

[3] F. Ahmadi-Abkenari, A. Selamat, *An architecture for a focused trend parallel Web crawler with the application of clickstream analysis,* „Information Sciences" 2012, vol. 184.

[4] Anuradha, A.K. Sharma, *Design of Hidden Web Search Engine,* „International Journal of Computer Applications" 2011, vol. 30 (9), Foundation of Computer Science, New York.

[5] S. Ceri, A. Bozzon, M. Brambilla, *The Anatomy of a Multidomain Search Infrastructure*, w: *Web Engineering*, red. S. Auer, O. Díaz, G. Papadopoulos, „Lecture Notes in Computer Science" 2011, vol. 6757, Springer, Berlin.

[6] I. Hernández, *A conceptual framework for efficient web crawling in virtual integration contexts*, w: *Proceedings of the 2011 International Conference on Web Information Systems and Mining – Volume Part II. WISM'11*, Springer-Verlag, Berlin–Heidelberg 2011.

## 2.2. Systemy przetwarzania języka naturalnego

Systemy wykorzystujące techniki przetwarzania języka naturalnego oraz ekstrakcję informacji często są zbudowane na zasadzie przetwarzania potokowego, intensywnie badanego już od lat 60. Przetwarzanie potokowe jest podzielone na następujące po sobie, wykonywane kolejno bloki przetwarzania[7]. W przypadku ekstrakcji informacji oraz przetwarzania języka naturalnego blokami takimi są np.: tokenizacja, normalizacja czy *stemming* i lematyzacja. Tokenizacja polega na wyodrębnieniu z tekstu poszczególnych słów, normalizacja na ujednoznacznieniu ich zapisu, a *stemming* i lematyzacja na sprowadzeniu tych słów do ich podstawowej formy[8]. Kolejnym etapem może być rozpoznawanie bytów nazwanych (ang. *named entity recognition*), które polega na rozpoznaniu w tekście wystąpień słów opisujących konkretne obiekty[9]. Często w tym celu są wykorzystywane słowniki lub ontologie zawierające pojęcia dziedzinowe, które mogą wystąpić w tekście.

Istnieje wiele rozwiązań komercyjnych oraz darmowych wspomagających przetwarzanie języka naturalnego[10]. Do najpopularniejszych z nich należy zaliczyć platformę GATE (General Architecture for Text Engineering)[11] oraz Apache UIMA (Unstructured Information Management Architecture)[12]. Obie platformy dostarczają rozwiązań wspomagających potokowe przetwarzanie języka naturalnego. Zarówno GATE, jak i UIMA umożliwiają stworzenie komponentów wykonujących kolejne fazy przetwarzania języka naturalnego oraz połączenie ich w potok przetwarzania.

Architektura wykorzystująca przetwarzanie potokowe jest także szeroko wykorzystywana przez projekty badawcze obejmujące swym zakresem przetwarzanie języka naturalnego oraz ekstrakcję informacji. Architektura taka została zaproponowana np. przez Mitchella i innych[13] dla systemu służącego do adnotacji raportów medycznych, Groovera i innych[14] dla systemu BioCreAtIvE II[15], służących do adnotacji tekstów poruszających zagadnienia z zakresu biologii, czy też Hongebooma i innych[16]

---

7 C. Ramamoorthy, H. Li, *Pipeline architecture,* „ACM Computing Surveys (CSUR)" 1977, vol. 9 (1).

8 Ch. Manning, P. Raghaven, H. Schütze, op.cit.

9 D. Nadeau, S. Sekine, *A survey of named entity recognition and classification,* „Lingvisticae Investigationes" 2007, vol. 30 (1).

10 Por.: http://en.wikipedia.org/wiki/List_of_natural_language_processing_toolkits.

11 http://gate.ac.uk/.

12 http://uima.apache.org/.

13 K. Mitchell, *Implementation and evaluation of a negation tagger in a pipeline based system for information extraction from pathology reports*, „Medinfo" 2004.

14 C. Grover, *Adapting a relation extraction pipeline for the BioCreAtIvE II task*, „Proceedings of the BioCreAtIvE II Workshop" 2007, vol. 2; Alex B., *Automating curation using a natural language processing pipeline*, „Genome Biology" 2008, vol. 9.

15 http://biocreative.sourceforge.net/.

16 A. Hogenboom, *Detecting Economic Events Using a Semantics-Based Pipeline*, w: *Database and Expert Systems Applications*, red. A. Hameurlain, „Lecture Notes in Computer Science" 2011, vol. 6860, Springer, Berlin–Heidelberg.

w systemie wykrywania wzmianek o zdarzeniach ekonomicznych w informacjach prasowych. We wszystkich tych systemach na proces przetwarzania języka naturalnego składają się fazy: tokenizacji, normalizacji, rozpoznawania bytów nazwanych oraz rozpoznawania relacji pomiędzy nimi.

# 3. Architektura rozwiązania

Architektura rozwiązania czerpie z wzorców znanych z obu przytoczonych wyżej klas systemów. Centralnymi punktami są dwa repozytoria – stron oraz zagrożeń. Zasileniem repozytorium stron zajmuje się hybrydowy robot, który jednocześnie aktualizuje bazę zagrożeń, wprowadzając informację o nowych dokumentach (stronach internetowych). Kolejne etapy wyodrębniają z dokumentów ogłoszenia sygnalizujące potencjalne zagrożenia (w terminologii przyjętej w projekcie dokument może zostać podzielony na dokumenty logiczne odpowiadające poszczególnym ogłoszeniom), ekstrahują informację o zagrożeniu (instancji profilu zagrożenia) i dokonują oceny jego stopnia.


**Rysunek 1. Ogólna architektura rozwiązania**

# 4. Główne komponenty funkcjonalne

## 4.1. Pozyskanie dokumentów

Proces pozyskiwania dokumentów jest realizowany w oparciu o podejście hybrydowe czerpiące wzorce zarówno z tradycyjnych robotów, jak i z robotów przeznaczonych do głębokiego Internetu. Robot jest odmianą robota ukierunkowanego – ogranicza się do zadanych źródeł i pozyskuje strony, stosując filtr umożliwiający wyodrębnienie stron prowadzących do ogłoszeń lub zawierających same ogłoszenia.

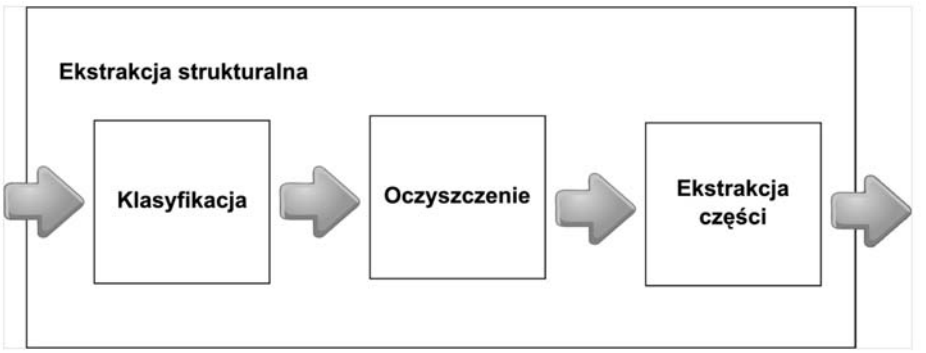
Wykorzystuje on równocześnie informację o strukturze hipertekstowej źródła, podobnie jak rozwiązania dla ukrytego Internetu, aby najpierw badać obszary źródła (portalu, forum) potencjalnie zawierające poszukiwane ogłoszenia.

## 4.2. Przetwarzanie dokumentów

Przetwarzanie dokumentów ma na celu zidentyfikowanie zagrożeń z nich wynikających i przygotowanie danych dla użytkownika końcowego. Obejmuje ono kolejno przeprowadzane fazy omówione w następnych sekcjach. W wyniku ekstrakcji strukturalnej jest wyodrębniona właściwa treść ogłoszeń, ekstrakcja leksykalna tworzy instancje profilu zagrożenia, tzn. wyodrębnia z ogłoszeń faktyczne zagrożenia wynikające z nielegalnego handlu lekami, a klasyfikacja ocenia ich stopień.

### 4.2.1. Ekstrakcja strukturalna

Zadaniem ekstrakcji strukturalnej jest przetworzenie strony pozyskanej ze źródła (dokumentu) do postaci fragmentów tekstu najbardziej interesujących ze względu na cel działania systemu. Jeżeli takim dokumentem jest strona zawierająca wpisy na forum lub pojedyncze ogłoszenie, to celem jest pobranie treści tego ogłoszenia, jego tytułu, osoby wystawiającej i danych o niej, odrzucona zostanie natomiast cała pozostała treść strony, nieistotna dla wykrywania zagrożeń. Ekstrakcja strukturalna przetwarza dokument na kilku etapach, pokazanych na rysunku 2.




**Rysunek 2. Ekstrakcja strukturalna**

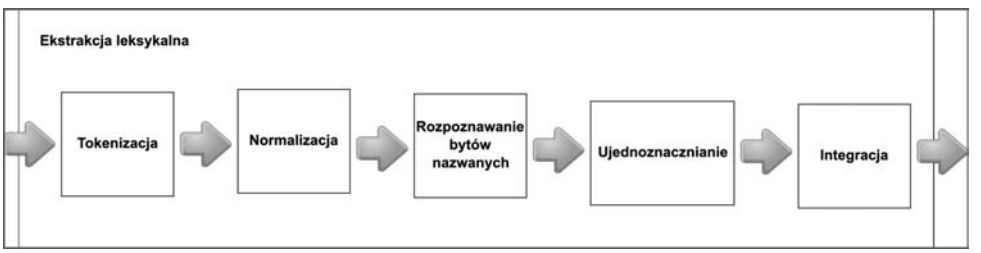
Pierwszym z nich jest rozpoznanie klasy dokumentu, czyli ustalenie, czy dokument należy do grupy interesujących dokumentów z danego źródła (tzn.: np. jest dokumentem z ogłoszeniem, a nie listą ogłoszeń). Etap ten jest wykonywany w oparciu o rozpoznanie charakterystycznego fragmentu adresu strony, charakterystycznego fragmentu tekstu w ramach dokumentu lub elementu jego struktury. Kolejnym etapem

jest przygotowanie dokumentu do przetwarzania przez oczyszczenie z błędów często spotykanych w kodzie HTML. Wykorzystywany jest do tego standardowy komponent HTML Tidy[17]. Wynikiem jego działania jest poprawny dokument XML. Na kolejnym etapie są stosowane przygotowane dla każdej klasy ręcznie arkusze XSLT, przy pomocy których jest dokonywana transformacja dokumentu do postaci jednego lub kilku dokumentów logicznych. Dokument logiczny odpowiada pojedynczemu wpisowi lub ogłoszeniu. Równocześnie w tych samych arkuszach XSLT są zdefiniowane odpowiednie wyrażenia w języku XPath stosowane do ekstrakcji wybranych fragmentów dokumentu, które zawierają istotne informacje dotyczące ogłoszenia (ogłoszeniodawca i jego dane kontaktowe, data opublikowania ogłoszenia, adres IP komputera, z którego nadano ogłoszenie itp.). Wynik ostatniego etapu, czyli ekstrakcji części, jest wprowadzany do bazy zagrożeń i stanowi podstawę do dalszej identyfikacji zagrożenia.

Głównym problemem przy realizacji ekstrakcji strukturalnej jest prawidłowe ustalenie klasy dokumentu. Umożliwia to niezawodne wykonanie ekstrakcji. Arkusze transformacji, za pomocą których jest dokonywana właściwa ekstrakcja, są powszechnie wykorzystywane w podobnych zadaniach i przy odpowiednim przygotowaniu wyrażeń ekstrahujących stanowią odporny na drobne zmiany w dokumentach mechanizm ekstrakcji[18].

### 4.2.2. Ekstrakcja leksykalna

Celem ekstrakcji leksykalnej jest zidentyfikowanie podstawowych informacji składających się na profil zagrożenia: danych ogłoszeniodawcy, rodzaju i ilości oferowanego leku, rodzaju oferty (kupno lub sprzedaż). Do tego celu jest stosowane podejście do przetwarzania języka naturalnego oparte na potoku wykonywanych na tekstach operacji. Przetwarzane teksty pochodzą z wcześniejszego etapu – ekstrakcji strukturalnej.




**Rysunek 3. Ekstrakcja leksykalna**

[17] www.w3.org/People/Raggett/tidy/.

[18] M. Kowalkiewicz, *Robust Web Content Extraction*, w: *15th International Conference on World Wide Web*, 2006.

Najpierw zostają one poddane tokenizacji, czyli podziałowi na słowa (rysunek 3). Głównym problemem, jaki należało rozwiązać na tym etapie, jest agramatyczność języka stosowanego w Internecie, co powoduje, że standardowe podejście do tokenizacji nie zdaje egzaminu. Po wyłonieniu słów następuje etap normalizacji niektórych z nich, np. dotyczących: dat, adresów e-mail, numerów telefonów, numerów kont i nazw miejscowych. Są one sprowadzane do postaci podstawowej lub ujednoliconego formatu. W połączeniu z normalizacją, a w niektórych przypadkach po niej, jest przeprowadzane rozpoznawanie bytów nazwanych, do których oprócz wyżej wspomnianych zaliczono także rodzaj oferty, ilość oferowanego leku, jego postać i formę (opakowanie, pojedyncze sztuki) oraz pseudonim ogłoszeniodawcy. W kolejnym etapie ekstrakcji leksykalnej porównuje się znalezione byty zidentyfikowane jako leki z bazą leków, tak aby rozpoznać, jaką najprawdopodobniej substancję wspomina dany dokument. Ostatni etap polega na zintegrowaniu oznaczonych w dokumencie informacji o bytach. Na tym etapie następuje skojarzenie np. adresu e-mail i adresu IP z osobą wspomnianą w dokumencie lub oferty z oferowaną substancją. Integracja pozwala wypełnić profil zagrożenia.

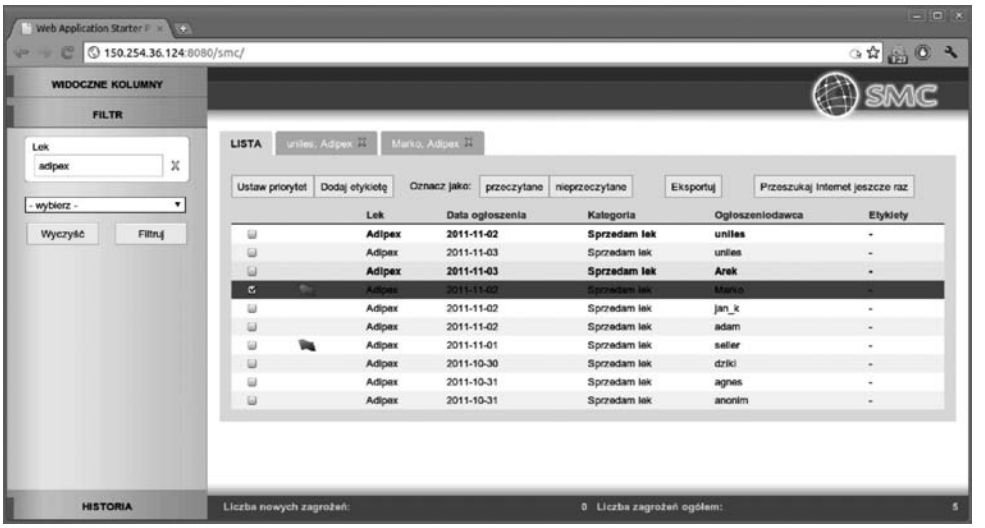
### 4.2.3. Klasyfikacja i ocena ryzyka

Kolejną fazą działania systemu jest heurystyczne oszacowanie ryzyka związanego z daną ofertą zawartą w dokumencie. Ryzyko to jest pochodną zidentyfikowanych przesłanek (rodzaj oferty, rodzaj substancji, jej ilość), jak również stopnia zaufania do wyekstrahowanych informacji, wynikającego ze skuteczności stosowanych metod (głównie ujednoznaczniania oraz rozpoznawania bytów nazwanych). Wykorzystywane są do tego wyliczone eksperymentalnie miary skuteczności stosowanych metod oraz dobrane heurystycznie parametry istotności poszczególnych parametrów. Zaliczenie zagrożenia do jednej z klas ryzyka kończy wypełnianie profilu zagrożenia.

## 4.3. Prezentacja

Ostatnią fazą procesu śledczego w informatyce śledczej jest prezentacja. W tworzonym systemie wykrywania zagrożeń w cyberprzestrzeni są prezentowane instancje profili zagrożeń, czyli w szczególności reprezentacje ogłoszeń dotyczących nielegalnej sprzedaży leków. Tworzony interfejs użytkownika jest aplikacją internetową, umożliwiającą przeglądanie listy profili zagrożeń (por. rysunek 4) oraz widoku szczegółowego profilu. W widoku szczegółowym profilu są prezentowane wszystkie informacje, które system pozyskał z treści ogłoszenia, natomiast w widoku listy użytkownik może wybrać, które elementy zagrożenia zostaną zaprezentowane (które kolumny mają być widoczne). Dodatkowo użytkownik ma dostęp do filtrowania, w ramach którego może określić, że widoczne będą tylko zagrożenia dotyczące

konkretnego leku, np. Adipeksu, albo ogłoszenia, które pojawiły się w określonym przedziale czasowym. System przechowuje historię przeglądania listy zagrożeń, dlatego możliwe jest także zastosowanie kryteriów filtrowania, z których użytkownik korzystał wcześniej.




**Rysunek 4. Widok interfejsu programu**

Podczas pracy z systemem użytkownik może oznaczyć dane zagrożenie definiowanymi przez niego etykietami lub jednym z trzech priorytetów (niski, średni, wysoki). Zagrożenia są także oznaczane jako przeczytane lub nieprzeczytane – analogicznie do popularnych klientów poczty elektronicznej. Przewidywane jest generowanie wykresów kontekstowych dotyczących, przykładowo, rozkładu leków występujących na obecnej liście zagrożeń oraz możliwość eksportowania danych z listy zagrożeń do formatu programu Microsoft Excel.

## 5. Infrastruktura badawcza

Istotnym i nietypowym aspektem przygotowywanego rozwiązania, który miał wpływ na architekturę, jest dwoistość systemu: ma to być z jednej strony zaawansowany prototyp możliwy do wykorzystania w środowisku produkcyjnym, z drugiej strony – prototyp umożliwiający przeprowadzenie eksperymentów nad różnymi metodami pozyskania stron, ekstrakcji i analizy treści. Dlatego też poszczególne komponenty przygotowano tak, aby ich podstawowa funkcjonalność była niezależna od innych komponentów i możliwa do uruchomienia w trybie eksperymentalnym

w kontrolowanym środowisku, w którym można analizować wyniki działania pojedynczych reguł, algorytmów czy metod. Równocześnie funkcjonalności te zostały opakowane tak, aby mogły stanowić komponenty architektoniczne całkowitego rozwiązania. Rolą opakowania jest m.in. zapewnienie interfejsu do kontrolowania danego komponentu (jego uruchomienia, monitorowania statusu, zamknięcia), zasilenie rdzennej funkcjonalności w dane na podstawie monitorowania stale uzupełnianej bazy zagrożeń i repozytorium dokumentów, jak również zapewnienie równoległości przetwarzania i izolacji poszczególnych wątków realizujących rdzenną funkcjonalność w odniesieniu do poszczególnych dokumentów czy wyodrębnionych ogłoszeń.

## 6. Podsumowanie

W ramach projektu Semantyczny Monitoring Cyberprzestrzeni tworzony jest system informatyczny monitorujący zdefiniowane źródła internetowe pod kątem nielegalnej sprzedaży leków. Zbudowanie takiego systemu związane jest z takimi zagadnieniami, jak: periodyczne odpytywanie określonych stron internetowych oraz ekstrakcja informacji z tych stron (w celu zbudowania instancji profilu zagrożenia), określenie stopnia zagrożenia oraz prezentacja zagrożeń ułatwiająca ich analizę. Wyodrębniono następujące komponenty systemu: monitor źródeł, ekstrakcja strukturalna, ekstrakcja leksykalna, klasyfikacja oraz graficzny interfejs użytkownika. Architektura tworzonego systemu rozszerza znane z literatury przykłady dotyczące przetwarzania języka naturalnego, dodając do nich komponenty odpowiedzialne za pozyskiwanie dokumentów, budowanie profilu oraz klasyfikację. Poszczególne komponenty odpowiadają kolejnym etapom procesu śledczego w informatyce śledczej.

## 7. Dalsze prace

Dalsze prace nad systemem obejmują rozwój metod rozpoznawania bytów nazwanych oraz opracowanie metod dodawania nowych źródeł. Konieczne jest także opracowanie metody oceny poziomu zagrożenia na podstawie informacji pojawiających się w konkretnej instancji profilu zagrożenia oraz informacji wywnioskowanych na podstawie innych instancji. W ramach interfejsu programu zostaną jeszcze zaimplementowane: funkcjonalność zaawansowanego filtrowania listy zagrożeń, generowanie wykresów kontekstowych oraz eksport zagrożeń do formatu Microsoft Excel. Przewidywane są także prace nad zastosowaniem tworzonego systemu do innych niż nielegalny obrót produktami leczniczymi domen przestępstw dokonywanych w Internecie.

## Literatura


1. Ahmadi-Abkenari F., Selamat A., *An architecture for a focused trend parallel Web crawler with the application of clickstream analysis*, „Information Sciences" 2012, vol. 184.
2. Alex B., *Automating curation using a natural language processing pipeline*, „Genome Biology" 2008, vol. 9.
3. Anuradha, Sharma A.K., *Design of Hidden Web Search Engine*, „International Journal of Computer Applications" 2011, vol. 30 (9), Foundation of Computer Science, New York.
4. Ashcroft J., *Electronic Crime Scene Investigation – A Guide for First Responders. Tech. rep. U.S. Department of Justice*, 2001.
5. Ceri S., Bozzon A., Brambilla M., *The Anatomy of a Multi-domain Search Infrastructure*, w: *Web Engineering*, red. S. Auer, O. Díaz, G. Papadopoulos, „Lecture Notes in Computer Science" 2011, vol. 6757, Springer, Berlin.
6. Dzikowski J., *Wykrywanie przestępczości z wykorzystaniem informacji ze źródeł internetowych*, praca magisterska – Uniwersytet Ekonomiczny w Poznaniu, Poznań 2010.
7. Grover C., *Adapting a relation extraction pipeline for the BioCreAtIvE II task*, „Proceedings of the BioCreAtIvE II Workshop" 2007, vol. 2.
8. Hernández I., *A conceptual framework for efficient web crawling in virtual integration contexts*, w: *Proceedings of the 2011 International Conference on Web Information Systems and Mining – Volume Part II. WISM'11*, Springer-Verlag, Berlin–Heidelberg 2011.
9. Hogenboom A., *Detecting Economic Events Using a Semantics-Based Pipeline,* w: *Database and Expert Systems Applications*, red. A. Hameurlain, „Lecture Notes in Computer Science" 2011, vol. 6860, Springer, Berlin–Heidelberg.
10. Kowalkiewicz M., *Robust Web Content Extraction*, w: *15th International Conference on World Wide Web*, New York 2006.
11. Manning Ch., Raghavan P., Schütze H., *Introduction to Information Retrieval,* Cambridge University Press, Cambridge 2008.
12. Mitchell K.J., *Implementation and evaluation of a negation tagger in a pipeline based system for information extraction from pathology reports*, „Medinfo" 2004.
13. Nadeau D., Sekine S., *A survey of named entity recognition and classification*, „Lingvisticae Investigationes" 2007, vol. 30 (1).
14. Pollitt M., *Applying traditional forensic taxonomy to digital forensics*, „IFIP Int. Conf. Digital Forensics" 2008, vol. 285, red. I. Ray, S. Shenoi, Springer.
15. Ramamoorthy C., Li H., *Pipeline architecture*, „ACM Computing Surveys (CSUR)" 1977, vol. 9 (1).
16. Solomon M., Broom N., Barrett D., *Computer Forensics JumpStart,* Alameda, CA, USA: SYBEX Inc., 2004.

# Summary

## Architecture of the Threat Detection System for Cyberspace

The article presents an architecture of the threat detection system targeted at potential crimes that manifest themselves in the textual internet sources. The solution is based on the approaches applied in the natural language processing systems, as well as information retrieval systems. The goal of the system is to retrieve the documents from the selected internet sources (forums, online classifieds portals), analyse them, identify the threat signals, and gather the largest possible amount of information about the threat. The architecture draws from the pipelined document processing approach.