|  |  |
| --- | --- |
| Politechnika Warszawska  Wydział Elektroniki i Technik Informacyjnych  Instytut Informatyki | Rok akademicki 2012/2013 |

Praca Dyplomowa Magisterska

Adrian Wiśniewski

Inteligentne wykrywanie reklam w sieci WWW na podstawie adresów URL

Opiekun pracy:

prof. dr hab. inż. Mieczysław Muraszkiewicz

|  |  |
| --- | --- |
| Ocena |  |
| Podpis przewodniczącego  Komisji Egzaminu Dyplomowego |  |

|  |  |
| --- | --- |
|  | Kierunek: Informatyka  Specjalność: Inżynieria Systemów Informatycznych  Data urodzenia: 24 grudnia 1987 r.  Data rozpoczęcia studiów: październik 2006 r. |

Życiorys

Urodziłem się 24 grudnia 1987 r. w Nowym Dworze Mazowieckim. Mieszkam w Wołominie, gdzie ukończyłem Sportową Szkołę Podstawową nr 5 im. C. K. Norwida i Sportowe Gimnazjum nr 3 im. Polskich Olimpijczyków. Następnie uczęszczałem do klasy o profilu matematyczno-fizycznym w IV Liceum Ogólnokształcącym im. Adama Mickiewicza w Warszawie. W październiku 2006 r. rozpocząłem studia na Politechnice Warszawskiej na Wydziale Elektroniki i Technik Informacyjnych na kierunku Informatyka.

Rok później zacząłem pracować jako programista aplikacji internetowych oraz mobilnych. Przez kilka lat wykonywałem różne zlecenia dla marek takich jak McDonald’s, Allegro, czy gazeta.pl. Mimo wielu sukcesów, postanowiłem – zgodnie ze swoimi marzeniami z dzieciństwa – tworzyć gry komputerowe, więc w sierpniu 2011 r. zatrudniłem się w firmie 11 bit studios S.A. Przez dwa lata – do chwili obecnej – pracowałem nad grą Anomaly 2, dodając do silnika gry możliwość rozgrywki wieloosobowej przez sieć. Po zakończeniu studiów zamierzam przeprowadzić się do Stanów Zjednoczonych i podjąć pracę w firmie Google, gdzie będę mógł dalej rozwijać się w dziedzinie sztucznej inteligencji.

Od 2006 roku należę do społeczności twórców gier komputerowych www.gamedev.pl, a w latach 2009 – 2010 pełniłem rolę administratora i członka zarządu Koła Naukowego Twórców Gier Komputerowych „Polygon”.

|  |  |
| --- | --- |
|  | Podpis studenta |

Egzamin dyplomowy

Złożył egzamin dyplomowy dn. 2011 r.

Z wynikiem

Ogólny wynik studiów

Dodatkowe wnioski i uwagi Komisji

Streszczenie

ITIL to obecnie najpopularniejsza metodyka zarządzania infrastrukturą informatyczną przedsiębiorstwa, zorientowana na zarządzanie usługami i należąca do rodziny ITSM. Praca zawiera opis jej struktury i procesów, ze szczególnym naciskiem położonym na scharakteryzowanie zarządzania komponentami usług i konfiguracją oraz zarządzania zmianą. Stworzony system jest aplikacją internetową napisaną w Javie 6 EE przy użyciu biblioteki Spring i posiada moduły do obsługi wyżej wymienionych procesów. Jego praktyczne zastosowania przedstawiono na przykładzie opisanej firmy. Praca omawia wszystkie etapy powstawania systemu od analizy wymagań, przez projekt i implementację do testowania. Ponadto jest uzupełnieniem i kontynuacją pracy inżynierskiej Piotra Kalańskiego, skoncentrowanej na obsłudze funkcji Service Desk — zarządzania incydentami i zarządzania problemami.

*Słowa kluczowe:*

ITIL, ITSM, Service Desk, Zarządzanie komponentami usług i konfiguracją, SACM, Zarządzanie zmianą

Abstract

ITIL is presently the most popular information technology management framework. It is service management oriented approach that belongs to ITSM family. This thesis includes description of ITIL’s structure and all processes, with special focus on Service Asset and Configuration Management (SACM) and Change Management. The new-created system is a web application written using Java 6 EE and Spring framework. It has modules, which support these processes. It’s practical applications are shown on sample business instance. Thesis describes all stages of development: requirement analysis, design, implementation and tests. Moreover it’s complementary to engineer thesis written by Piotr Kalański, which focuses on Service Desk function, especially incident management and problem management.

*Keywords:*

ITIL, ITSM, Service Desk, Service Asset and Configuration Management, SACM, Change Management

Spis treści

[Rozdział 1. Wstęp 5](#_Toc356921285)

[1.1. Cel pracy 5](#_Toc356921286)

[1.2. Zakres 6](#_Toc356921287)

[1.3. Powiązane prace naukowe i publikacje 6](#_Toc356921288)

[Rozdział 2. Filtrowanie reklam w sieci WWW 7](#_Toc356921289)

[2.1. Po czym rozpoznać reklamę? 8](#_Toc356921290)

[2.2. Obecne rozwiązania 10](#_Toc356921291)

[2.3. Metody inteligentne 10](#_Toc356921292)

[Rozdział 3. Inteligentny system wykrywania 11](#_Toc356921293)

[3.1. Pozyskiwanie danych 11](#_Toc356921294)

[3.2. Przetwarzanie wstępne 11](#_Toc356921295)

[3.3. Dobór klasyfikatorów 11](#_Toc356921296)

[3.4. Trenowanie 11](#_Toc356921297)

[3.5. Agent produkcyjny 11](#_Toc356921298)

[Rozdział 4. Implementacja 12](#_Toc356921299)

[Rozdział 5. Wyniki 13](#_Toc356921300)

[5.1. Zebrane dane 13](#_Toc356921301)

[5.2. Skuteczność systemu 13](#_Toc356921302)

[Rozdział 6. Podsumowanie 14](#_Toc356921303)

[Bibliografia 15](#_Toc356921304)

[Spis ilustracji 18](#_Toc356921305)

[Dodatek A. Zawartość płyty CD 19](#_Toc356921306)

[Dodatek B. Konfiguracja i uruchamianie systemu 20](#_Toc356921307)

# Wstęp

Wraz z rozwojem Internetu i sieci WWW w bardzo szybkim tempie wzrasta ilość danych przesyłanych łączami telekomunikacyjnymi. Jednak oprócz zwiększania się ilości treści pożądanych przez użytkownika, można zaobserwować także znaczący wzrost liczby reklam. Agencje wykorzystując ciągły przyrost szybkości transferu, umieszczają w sieci coraz obszerniejsze materiały promocyjne: grafiki, animacje, a nawet całe klipy wideo zawierające dźwięk. Im więcej takich elementów zawiera strona internetowa, tym dłużej użytkownik musi czekać na jej załadowanie, a pliki reklam bardzo często mają znaczący udział w ilości danych pobranych przez przeglądarkę klienta. Dzieje się tak ponieważ – w odróżnieniu od pozostałych grafik stanowiących stałe elementy układu graficznego strony – reklamy zwykle nie są zapisywane w pamięci podręcznej przeglądarki. Przy przechodzeniu między kolejnymi stronami portalu efektywnie pobieramy pojedynczy plik html i wiele materiałów reklamowych. Nawet małe reklamy w dużych ilościach potrafią spowolnić proces ładowania strony internetowej, ponieważ zmuszają przeglądarkę do wykonania wielu zapytań do wielu różnych serwerów. Sytuacja intensyfikuje się szczególnie w sieciach mobilnych, w których opłaty są naliczane w zależności od objętości transferowanych danych. Użytkownik w tym momencie traci podwójnie, ponieważ nie dość, że zwykle nie ma ochoty oglądać reklam, to jeszcze musi za tą wątpliwą przyjemność płacić.

Z wyżej wymienionych powodów narasta potrzeba oddzielenia rzeczywistej treści stron internetowych od tej niepożądanej, podobnie jak nie tak dawno temu zaczęto oddzielać spam od prawdziwych wiadomości poczty elektronicznej.

## Cel pracy

Stworzenie systemu do wykrywania reklam w sieci WWW opartego o metody uczenia maszynowego, który będzie w stanie odróżnić reklamy od treści w sposób nie wymagający ingerencji użytkownika.

Istniejące obecnie rozwiązania pozwalają na filtrowanie reklam, jednak są one oparte na czarnych listach serwerów reklamowych, do których wysyłanie żądań HTTP jest blokowane na poziomie przeglądarki internetowej. Mimo swojej dużej skuteczności aplikacje te wymagają częstego i manualnego uaktualniania list adresów. Pojawia się tutaj szerokie pole do zastosowania algorytmów uczenia maszynowego oraz eksploracji danych. Metody te umożliwiają zautomatyzowanie całego procesu, zwiększenie skuteczności filtracji oraz przede wszystkim minimalizację kosztów utrzymania aplikacji.

## Zakres

System powinien być nieinwazyjny i współdziałać ze wszystkimi przeglądarkami internetowymi. Można to osiągnąć poprzez zbudowanie odpowiedniego pośredniczącego (ang. proxy) między przeglądarką, a siecią WWW. Serwer ten może filtrować ruch i odrzucać zapytania HTTP dotyczące materiałów reklamowych.

System powinien pozwolić na ocenę i porównanie skuteczności wielu różnych metod uczenia maszynowego takich jak: naiwny klasyfikator Bayesa, sieć Bayesa, klasyfikator k-najbliższych sąsiadów, maszyna wektorów nośnych, sieć neuronowa, drzewo decyzyjne, las drzew losowych, meta-klasyfikator AdaBoost oraz inne. Ponadto musi istnieć możliwość podania różnych zestawów parametrów dla wymienionych metod.

## Powiązane prace naukowe i publikacje

Niniejsza praca powstała w ramach projektu SmartAdBlocker prowadzonego przez Politechnikę Warszawską na zlecenie T-Mobile. Projekt był prowadzony pod przewodnictwem prof. dr hab. inż. Mieczysława Muraszkiewicza w czteroosobowym zespole:

* Piotr Szczepański (kierownik projektu)
* Adrian Wiśniewski
* Michał Januszewski
* Krzysztof Kamiński

Moim zadaniem było opracowanie i implementacja architektury systemu, co zostało opisane w tej pracy naukowej. Powstały jeszcze dwie prace naukowe:

* O proxy – Januszewski
* O wynikach – Kamiński

Ponadto powstała publikacja naukowa:

* An Automated Framework with Application to Study URL Based Online Advertisements Detection

# Filtrowanie reklam w sieci WWW

Poruszając się po sieci WWW wykonujemy szereg zapytań protokołu HTTP[[1]](#footnote-1). Protokół ten pozwala zażądać od wskazanego serwera przesłania zasobu znajdującego się pod podanym adresem URL[[2]](#footnote-2). Najczęściej zasobem tym jest strona internetowa, będąca plikiem tekstowym sformatowanym zgodnie z językiem HTML[[3]](#footnote-3). Plik ten zawiera opis układu strony: definicje różnych elementów, ich wzajemne położenie oraz wskazówki dotyczące ich formatowania. Zasadniczymi elementami pojawiającymi się w tego typu dokumencie są bloki tekstowe zawierające treść, hiperłącza oraz odwołania do innych zasobów takich jak: obrazy, animacje SWF[[4]](#footnote-4), skrypty Javascript i arkusze stylów CSS[[5]](#footnote-5).

Można wyróżnić dwa główne typy reklam. Pierwszym z nich są wyskakujące okna (ang. pop-up). Są to specjalnie przygotowane strony internetowe zawierające jedynie reklamy, które pojawiają się zaraz po wejściu na inną stronę. Jest to najbardziej irytujący typ reklam, ponieważ zasłania całą treść oraz zmusza użytkownika do przerwania bieżącej czynności i zamknięcia wyskakującego okienka. Początkowo takie reklamy były bardzo popularne, jednak w związku ze zbyt dużą natarczywością reklamodawców, w przeglądarkach internetowych wprowadzono mechanizmu blokowania okienek wyskakujących, co spowodowało znaczny spadek ilości reklam dystrybuowanych tą metodą. Istnieje także odmiana wyświetlająca okienko pod oknem z treścią (ang. pop-under).

Drugim typem są reklamy osadzone, umieszczane obok treści, jako elementy strony internetowej. Zazwyczaj przybierają formę obrazów lub animacji SWF, ale mogą być to także reklamy tekstowe. Są znacznie mniej inwazyjne, jednak nadal zaciemniają treść, ponieważ znajdują się w najbardziej widocznych i eksponowanych miejscach. Reklamy tego typu występują w wielu kształtach i rozmiarach. Od tapet umieszczanych w tle („watermark”), przez obrazki („button” - 120x60 px, „box” - 125x125 px, „baner” – 400x50 px) do reklam zakrywających prezentowaną treść („toplayer”, „interstitial”). Często zawierają animacje, których ruch dodatkowo odwraca uwagę użytkownika od czytanej treści. Ponadto niektóre starają się upodobnić do okienek systemu operacyjnego, co dodatkowo dezorientuje i wprowadza w błąd.

## Po czym rozpoznać reklamę?

Aby rozpoznać reklamę, musimy określić zbiór cech, które odróżnią ją od treści strony internetowej. Celem budowanego systemu jest minimalizacja danych pobieranych przez sieć, dlatego należy skupić się na cechach, które nie wymagają wysyłania zapytania do serwera.

W przypadku reklam z grupy wyskakujących okienek, jedyną informacją jaką posiadamy jest adres URL żądanej strony. Wyświetlenie tych okienek jest inicjowane najczęściej przez skrypty osadzone w stronie internetowej, co znacząco utrudnia rozpoznanie kontekstu, w jakim dane żądanie zostało zgłoszone. Bez znacznej ingerencji w przeglądarkę nie będziemy posiadać informacji, czy okienko otworzyło się samo, czy w związku z akcją użytkownika np. kliknięciem łącza otwierającego galerię zdjęć.

W przypadku reklam osadzonych znamy dodatkowo kontekst elementu: jego rozmiar, położenie oraz szereg innych atrybutów. Rozmiar jest bardzo obiecującą cechą, ponieważ w przypadku reklam jest zwykle ustandaryzowany – agencja tworzy co najwyżej kilka wersji reklamy i oczekuje, że na wszystkich stronach internetowych, na których reklamy te będą umieszczane, będą one wyglądały dobrze. Natomiast administrator strony internetowej umieszcza reklamy od różnych agencji i nie zmienia przy tym układu swojej witryny. Możemy wykorzystać także informację dotyczącą położenia elementu względem innych elementów strony. Jeżeli mamy do czynienia z dużym poziomym obrazkiem u góry strony, istnieje spora szansa, że jest to baner reklamowy. Podobnie z obrazkami, które przykrywają tekst na stronie. Bardzo często nazwy elementów i klas stylów zdradzają przeznaczenie elementu np.:



Listing : Fragment strony http://www.wp.pl[[6]](#footnote-6) -- reklamy znajdują się w blokach div oznaczonych jako adv

Mimo tego, analiza kontekstu elementu na stronie internetowej jest zadaniem dosyć ciężkim i czasochłonnym, w przeciwieństwie do analizy adresów URL, które są w znacznie krótsze i prostsze. Adresy te zawierają wystarczająco informacji, aby rozróżnić reklamy od treści. Korzystając z nich odrzucamy na wstępie możliwość filtrowania osadzonych reklam tekstowych, które z racji swojej natury nie są oddzielnym zasobem i nie posiadają adresu URL. Reklamy te pobieramy razem ze stroną internetową, dlatego na ich usuwaniu nie zaoszczędzimy transferu. Na poniższym listingu możemy bez trudu zobaczyć jak dużo reklam pobiera przeglądarka, ponieważ angielskie słowo „ads” oznaczające „reklamy” zdradza zawartość zasobów:



Listing : Przykładowe adresy URL zasobów pobranych przez przeglądarkę podczas sesji

Kolejne cechy możemy uzyskać dopiero po wysłaniu żądania do serwera i odebraniu zasobu. Pozwoli nam to uzyskać informacje o rozmiarze i typie MIME[[7]](#footnote-7) pliku oraz innych nagłówkach HTTP wysłanych przez serwer (w szczególności dotyczących kontroli pamięci podręcznej). Jest to ogromna ilość bardzo użytecznych informacji. Rozmiary większości reklam są ustandaryzowane i nie przekraczają dozwolonej dla danego formatu liczby kilobajtów. Także niewielka liczba typów plików jest używana jako reklamy.

Ponadto da nam dostęp do samej treści zasobu, z której można wydobyć kolejne cechy. W przypadku multimediów można nawet zastosować metody rozpoznawania obrazów i głosu. Taka analiza wymagałaby jednak znaczących nakładów mocy obliczeniowej i obecnie nie jest możliwa ze względów praktycznych, ponieważ powodowałaby olbrzymie opóźnienia w czasie wyświetlania strony.

## Obecne rozwiązania

Istnieje bardzo dużo programów i wtyczek do przeglądarek pozwalających na filtrowanie reklam. Te najprostsze modyfikują plik hosts, który w systemie Windows jest odpowiedzialny za tłumaczenie nazw hostów na adresy IP. Zmiana polega na przekierowaniu żądań do znanych serwerów reklamowych do niepoprawnego adresu IP, co uniemożliwia pobranie plików z reklamami. Rozwiązanie to wymaga podania całej nazwy hosta i nie adaptuje się w żadnym stopniu do zmieniających się warunków.

Jednak najpopularniejsze rozwiązania, takie jak Adblock Plus[[8]](#footnote-8) czy AdBlock[[9]](#footnote-9), opierają się o listy wyrażeń regularnych. Poszczególne wyrażenia – zwane filtrami – są porównywane z adresami URL żądań. Większość filtrów jest stosowana jako czarna lista – żądanie jest od razu odrzucane, jeżeli adres do nich pasuje. Istnieje także zbiór filtrów działających jako biała lista, pozwalających na bezpieczne pobranie zasobu. Listy wyrażeń są tworzone i utrzymywane najczęściej niezależnie od programów. Najpopularniejszą jest EasyList[[10]](#footnote-10), która jest używana przez ponad 12 milionów użytkowników[[11]](#footnote-11). Zawiera ona li

## Metody inteligentne

# Inteligentny system wykrywania

Strona 10, 11

## Pozyskiwanie danych

Strona 12, 13

## Przetwarzanie wstępne

Strona 14, 15, 16, 17, 18

## Dobór klasyfikatorów

Strona 19, 20, 21, 22

## Trenowanie

Strona 23

## Agent produkcyjny

Strona 24

# Implementacja

5 stron

# Wyniki

5 stron

## Zebrane dane

## Skuteczność systemu

# Podsumowanie

2 strony

Bibliografia

1. **Kalański Piotr.** *System Service Desk zgodny z zaleceniami ITIL.* 2010.

2. **OGC.** ITILv3 Glossary Polish. *ITIL Official Site.* [Online] 2008. [Zacytowano: 10 Grudnia 2010.] http://www.itil-officialsite.com/nmsruntime/saveasdialog.aspx?lID=921&sID=242.

3. —. *Service Transition Book (ITIL).* Londyn : The Stationery Office, 2007.

4. **Knowledge Transfer.** Słownik pojęć ITIL. [Online] [Zacytowano: 20 Lipiec 2011.] http://www.knowledgetransfer.net/dictionary/ITIL/en/IT\_Service\_Management.htm.

5. —. Słownik pojęć ITIL - hasło Function. [Online] [Zacytowano: 20 Lipiec 2011.] http://www.knowledgetransfer.net/dictionary/ITIL/en/Function.htm.

6. **Global Journals Inc.** *Global Journal of Computer Science and Technology.* [Online] Grudzień 2010. [Zacytowano: 18 Lipiec 2011.] http://computerresearch.org/stpr/index.php/gjcst/article/download/465/424.

7. ITIL Service Management - A Brief History of ITIL. *ITIL Service Management.* [Online] 18 Lipca 2011. [Zacytowano: 18 Lipca 2011.] http://itservicemngmt.blogspot.com/2007/09/brief-history-of-itil.html.

8. IT Life - Korzenie ITIL. *IT Life.* [Online] [Zacytowano: 18 Lipiec 2011.] http://itsm.itlife.pl/content/view/10012/57/.

9. **Ratcliffe David.** Pink Elephant - IT Service Management 2007. *Pink Elephant.* [Online] [Zacytowano: 18 Lipiec 2011.] http://www.pinkelephant.com/articles/MicrosoftPowerPointDavidRatcliffePinkPerspectiveBM.pdf.

10. *actis-ingenierie.* [Online] http://www.actis-ingenierie.com.

11. Projekt ITIL Refresh. *ITLife.pl.* [Online] [Zacytowano: 19 Lipiec 2011.] http://itsm.itlife.pl/content/view/10071/138/.

12. **OGC.** Scope and Development Plan: ITIL® V3 Update. [Online] February 2010. [Zacytowano: 21 Lipiec 2011.] http://www.best-management-practice.com/gempdf/Scope\_and\_Development\_Plan\_ITIL\_V3\_Update.pdf.

13. IT Life - Co to jest ITIL? *IT Life.* [Online] [Zacytowano: 7 Lipiec 2011.] http://itsm.itlife.pl/.

14. IT frameworks wiki - Hasło ITIL. *IT frameworks wiki.* [Online] [Zacytowano: 7 Lipiec 2011.] http://www.itframeworks.org/wiki/Information\_Technology\_Infrastructure\_Library.

15. **OGC.** *Service Strategy Book (ITIL).* brak miejsca : The Stationery Office, 2007.

16. **itSMF Ltd.** *An Introductory Overview of ITIL® V3.* brak miejsca : The UK Chapter of the itSMF, 2007.

17. **OGC.** *Service Operation Book (ITIL).* brak miejsca : The Stationery Office, 2007.

18. —. *Continual Service Improvement Book (ITIL).* brak miejsca : The Stationery Office, 2007.

19. **Knowledge Transfer.** Słownik pojęć ITIL. [Online] [Zacytowano: 20 Lipiec 2011.] http://www.knowledgetransfer.net/dictionary/ITIL/en/Configuration\_Item.htm.

20. **Wikipedia.** Configuration management - Wikipedia, the free encyclopedia. [Online] [Zacytowano: 20 Lipiec 2011.] http://en.wikipedia.org/wiki/File:ConfiurationActivityModel.png.

21. **Knowledge Transfer.** Słownik pojęć ITIL. [Online] [Zacytowano: 20 Lipiec 2011.] http://www.knowledgetransfer.net/dictionary/ITIL/en/Change.htm.

22. **Sante Tom i Jeroen Ermers.** *TOGAF™ 9 and ITIL® V3 Two Frameworks Whitepaper.* 2009.

23. **Carter Greg.** How ITIL and TOGAF Help Bridge the Chasm Between Business and IT. [Online] 10 Grudzień 2009. http://www.ctoedge.com/content/how-itil-and-togaf-help-bridge-chasm-between-business-and-it.

24. **Microsoft.** In-Depth View of MOF Process Model Quadrants. [Online] http://technet.microsoft.com/en-us/library/cc539253.aspx.

25. **Van Bon Jan, i inni.** Cross-Reference ITIL® V3 and MOF 4.0. [Online] 2009. http://www.best-management-practice.com/gempdf/Cross\_Ref\_ITILV3\_MOF4.pdf.

26. **Gamma Erich , i inni.** *Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku.* brak miejsca : Wydawnictwa Naukowo Techniczne, 2005.

27. **Fowler Henry.** *Architektura systemów zarządzania przedsiębiorstwem. Wzorce projektowe.* brak miejsca : Helion, 2005.

28. **Stochmiałek Michał.** Wprowadzenie do programowania aspektowego. [Online] http://stochmialek.pl/papers/aop-intro-seminar.pdf.

29. **Collins Dan.** Generic Data Modeling. [Online] http://www.dama-nj.org/presentations/Kalido\_Generic\_Data\_Modeling.pdf.

30. **JBoss.** Dokumentacja Hibernate. [Online] http://docs.jboss.org/hibernate/core/3.3/reference/en/html/inheritance.html.

31. **Keogh Patric.** The Value of Cloud Computing to ITSM. [Online] [Zacytowano: 14 Lipiec 2011.] http://www.slideshare.net/patrick\_keogh/the-value-of-cloud-computing-to-itsm.

32. **Scott Noel.** Case Study: Using ITIL® and PRINCE2™ Together. [Online] 2010. http://www.best-management-practice.com/gempdf/Using\_ITIL\_and\_PRINCE2\_Together\_August\_2010.pdf.

Spis ilustracji

Ilustracja 2.1. Obszary ITIL v2 [11] 17

Ilustracja 2.2. Cykl życia usługi w ITIL [14] 18

Ilustracja 2.3. Procesy ITIL pogrupowane wg hierarchii zarządzania [15] 19

Ilustracja 2.4 Model czynności zarządzania konfiguracją [21] 29

Ilustracja 3.1. Przykładowe elementy konfiguracji firmy 42

Ilustracja 4.1 Warstwy systemu 54

Ilustracja 4.2 Schemat podziału warstwy aplikacji na podwarstwy 54

Ilustracja 4.3 Schemat budowy warstwy prezentacji 56

Ilustracja 4.4 Diagram pakietów systemu Service Desk 57

Ilustracja 5.1 Tabele przechowujące dane o uniwersalnych elementach konfiguracji 65

Ilustracja 5.2 Tabele przechowujące dane pozostałych typów elementów konfiguracji 68

Ilustracja 6.1 Tabele przechowujące dane wniosków zmian 74

Ilustracja 6.2 Tabele przechowujące dane kategorii zmian i organów autoryzacji 75

Ilustracja 6.3 Tabele przechowujące dane historyczne wniosków zmian 75

Ilustracja 7.1 Klasy elementów konfiguracji firmy RedHost 77

Ilustracja 7.2 Szczegóły przykładowego elementu konfiguracji 79

Ilustracja 7.3 Lista przykładowych elementów konfiguracji 80

Ilustracja 7.4 Szczegóły złożonego wniosku 82

Ilustracja 7.5 Historia rozpatrzonego i wdrożonego wniosku 83

Ilustracja 7.6 Panel akcji wniosku o zmianę 84

Ilustracja 7.7 Powiadomienia generowane podczas przetwarzania wniosku 85

Ilustracja 7.8 Kategorie wniosków zmian zgodne z planem zarządzania zmianą 86

Zawartość płyty CD

Konfiguracja i uruchamianie systemu

1. ang. Hypertext Transfer Protocol – protokół komunikacji między przeglądarką, a serwerem WWW [↑](#footnote-ref-1)
2. ang. Uniform Resource Locator – ujednolicony zapis adresu zasobu [↑](#footnote-ref-2)
3. ang. HyperText Markup Language – język zapisu dokumentów w sieci WWW [↑](#footnote-ref-3)
4. Plik z animacją odtwarzaną przez Adobe Flash Player [↑](#footnote-ref-4)
5. ang. Cascading Style Sheets – arkusze styli z opisem wyglądu elementów (rozmiar, kolor itp.) [↑](#footnote-ref-5)
6. Pobrano dnia 21.05.2013 o godzinie 15:34 przy użyciu przeglądarki Firefox [↑](#footnote-ref-6)
7. ang. Multipurpose Internet Mail Extensions – identyfikator formatu przesyłanego pliku [↑](#footnote-ref-7)
8. http://adblockplus.org/en/firefox [↑](#footnote-ref-8)
9. http://code.google.com/p/adblockforchrome/ [↑](#footnote-ref-9)
10. http://easylist.adblockplus.org/en/ [↑](#footnote-ref-10)
11. https://easylist.adblockplus.org/blog/2011/09/01/easylist-statistics:-august-2011 [↑](#footnote-ref-11)