dr inż. Marian Wrzesień,
Piotr Ryszawa,
Przemysłowy Instytut Automatyki i Pomiarów

SYSTEM ZARZĄDZANIA DOKUMENTAMI PROJEKTU PROTEUS ZAIMPLEMENTOWANY W OpenKM

Zaprezentowano implementację systemu zarządzania dokumentami (DMS) przeznaczonego do nadzoru obiegu dokumentów generowanych podczas realizacji projektu PROTEUS. Jako podstawę do realizacji systemu wykorzystano oprogramowanie OpenKM Community instalowane pod systemem operacyjnym Windows Server 2003. Omówiono proces instalacji oraz konfiguracji serwera JBoss aplikacji Java. System zarządzania dokumentami DMS umożliwia dostęp rozproszonego zespołu konsorcjantów do zasobów informatycznych posadowionych w serwerze JBoss. Webowy interfejs użytkownika, zgodny ze specyfikacją Web 2.0, powoduje, że nie jest wymagane instalowanie lokalnego oprogramowania klienckiego. Podkreślono bezpieczeństwo transferu danych pomiędzy serwerem i użytkownikami osiągnięte przez zaimplementowanie protokołu ssl wspomaganego wygenerowanym dla potrzeb systemu certyfikatem bezpieczeństwa oraz restrykcje przy nadawaniu praw dostępu. Omówiono wybrane elementy funkcjonalności systemu tak z pozycji użytkownika jak i administratora.

THE PROTEUS PROJECT DOCUMENTS' MANAGEMENT SYSTEM IMPLEMENTED OVER OpenKM

The implementation of the Document Management System (DMS) dedicated for the supervising of the documents' route, which are generated while realizing PROTEUS project, is presented. The base software for the system is the OpenKM Community installed under the Windows Server 2003 OS. The installation steps as well as the configuration process were discussed. Document Management System offers remote access to the IT resources, which are located in the JBoss server, dedicated for the scattered consortium team. The compliance of the web user interface with the Web2.0 specification causes there is no need to put in any local client software. It is emphasized, that the server-client data transfer security could be achieved thanks to ssl protocol implementation supported with the certificate which was created for the security reason and the restrictions while involving the users access. Some selected systems' features intended for users as well as for the administrator use were discussed.

1. WSTĘP

System Zarządzania Dokumentami (DMS – *Dokument Management System*) w postaci cyfrowej pozwala na wydajne zarządzanie dużymi zbiorami informacji, ich wprowadzanie, modyfikacje, przeglądanie, przeszukiwanie, sterowanie hierarchicznym dostępem do zasobów DMS, a co najważniejsze udostępnianie tak lokalnie, jak i w środowisku rozproszonym użytkowników, z zachowaniem bezpieczeństwa transferu danych.

Efektem stosowania DMS jest uzyskanie dużej wydajności podczas przetwarzania danych, bezpieczeństwa ich przechowywania i transferu, łatwego w użyciu, w polskiej wersji językowej, narzędzia niezbędnego podczas realizacji wspólnego zadania przez wielu współwykonawców zlokalizowanych w rozproszonych siedzibach.

2. SYSTEM DMS PIAP W OPARCIU O OPENKM 3.0 COMMUNITY

System DMS PIAP został wdrożony dla potrzeb obsługi unijnego projektu PROTEUS, przy którym współpracuje ze sobą wiele instytucji (konsorcjantów) realizujących różnorakie zadania składające się na jedno wspólne rozwiązanie finalne. Do wspomagania tego przedsięwzięcia, w zakresie koordynacji obiegu dokumentacji pomiędzy tymi instytucjami, zastosowano System Zarządzania Dokumentami z wykorzystaniem pakietu oprogramowania. OpenKM 3.0 Community. O wyborze tego produktu zadecydowały wcześniej wymienione cechy użytkowe.

2.1. Czym jest OpenKM?

OpenKM jest systemem zarządzania dokumentami, wyposażonym w webowy interfejs użytkownika, nie wymagający instalowania lokalnego oprogramowania klienckiego. Oznacza to, że system ten umożliwia współpracę tak grup pracowniczych zlokalizowanych w tej samej siedzibie, jak i zespołu rozproszonego – komunikującego się przy pomocy sieci Internetowej. Ta funkcjonalność decyduje o technologicznej atrakcyjności tego rozwiązania. Dodatkowo system ten umożliwia wybór wersji językowej oraz udostępnia zdalne administrowanie systemem.

W OpenKM znajdują zastosowanie takie technologie open source, jak J2EE, JBoss, Ajax web (GWT) oraz Jackrabbit (lucene). Technologia J2EE (Java Platform, Enterprise Edition) definiuje standard tworzenia aplikacji w języku programowania Java oparty o wielowarstwową architekturę komponentową. Komponenty są zwykle osadzane na serwerze aplikacyjnym obsługującym Java Enterprise.

JBoss pełni funkcję serwera aplikacji w Javie na licencji LGPL, na bazie technologii Enterprise JavaBeans (EJB). Serwer ten został napisany w całości w języku Java, dzięki czemu JBoss jest dostępny dla niemal wszystkich platform. JBoss implementuje pełen zestaw usług J2EE.

Istotą technologii Enterprise JavaBeans jest tworzenie komponentów (ziarna) EJB, które są osadzane na serwerze aplikacji JBoss, skąd są udostępniane lokalnie lub zdalnie przez protokół RMI (zdalne wywoływanie metod obiektów) poprzez IIOP (Internet Inter-ORB Protocol). Trzy główne ziarna EJB to:

- Sesyjne EJB używane do umieszczania w nich logiki aplikacji - czyli kodu, który przetwarza dane,
- Encyjne EJB reprezentujące w sposób obiektowy dane (np. dostarczają obiektowego spojrzenia na relacyjną bazę danych).
- Ziarna sterowane komunikatami wykorzystywane w przetwarzaniu asynchronicznym i w zaawansowanych modelach współpracy oprogramowania. Np. model abonent-dostawca: bean rejestruje się jako dostawca pewnej usługi, klienci mogą zarejestrować się jako abonenci.

W pakiecie instalacyjnym, który można pobrać ze strony producenta **OpenKM** dostępne jest oprogramowanie wraz z serwerem JBoss zapewniającym współpracę z aplikacją. Domyślnie OpenKM współpracuje z bazą danych HyperSQL (freeware). Opcję tę można zmienić w drodze bezpośredniej modyfikacji plików konfiguracyjnych.

Do programowania w języku Java - podczas implementacji systemu – wykorzystywane jest darmowe oprogramowanie firmy Sun Microsystems **Java Development Kit** (**JDK**) – tu, w wersji dla Microsoft Windows.

2.2. Implementacja sytemu

System OpenKM został zaimplementowany w serwerze wyposażonym w system operacyjny Windows Server 2003. Wymagało to zainstalowania systemu, jego skonfigurowania oraz wygenerowanie i zaimplementowanie certyfikatu bezpieczeństwa.

2.2.1. Instalacja systemu w serwerze Windows 2003

- Pobranie (ze strony http://www.sun.com) i instalacja Java Developer Kit (JDK) 6.0
- Pobranie (ze strony http://www.openkm.com) i instalacja systemu OpenKm ver 3.0 community - plik OpenKM-3.0-RC1_JBoss-4.2.2.GA.zip zawierający serwer aplikacji JBoss

Domyślnym adresem systemu OpenKM jest http://localhost:8080/OpenKM/.

2.2.2. Konfiguracja systemu w serwerze Windows 2003

Konfiguracja systemu obejmuje:

- zdefiniowanie bezpiecznego połączenia https na bazie certyfikatu ssl z wykorzystaniem portu 7443. Uwzględniający te wymogi plik konfiguracyjny **serwer.xml** zamieszczono w załączniku **A**.
- skonfigurowanie usługi e-mail z wykorzystaniem stosowanego w PIAP serwera pocztowego ni.piap.pl. Uwzględniający to zdefiniowanie plik konfiguracyjny **mail-service.xml** zamieszczono w załączniku **B**
- skonfigurowanie parametrów pracy Systemu Zarządzania Dokumentami. Plik konfiguracyjny OpenKM.cfg zamieszczono w załączniku **C**.

2.2.3. Generowanie certyfikatu

Do wygenerowania certyfikatu zastosowano narzędzie keytool dostarczane z pakietem JDK6. W tym celu z konsoli, w katalogu uruchomieniowym C:\Program Files\Java\jdk\bin\") należy podać polecenie :

keytool –genkey –v -alias serverX –dname "CN=serverX**,OU=IT,O=JPC,C=GB" -keypass password -keystore serverX.jks -storepass password -keyalg "RSA" -sigalg "MD5withRSA" -keysize 2048 -validity 3650**

Wygenerowane Certyfikaty Java znajdują się w katalogu uruchomieniowym narzędzia keytool. Docelowo, w systemie Windows 2003, certyfikaty powinny być przechowywane w pliku:

„C:\Program Files\Java\jre6\lib\security\cacerts"

W celu zaimportowania certyfikatu ssl do zbioru certyfikatów Java w Serwerze Windows 2003 należy uruchomić polecenie:

keytool –import -alias serverX -file serverX.cer -keystore "C:\Program Files\Java\jre6\lib\security\cacerts"

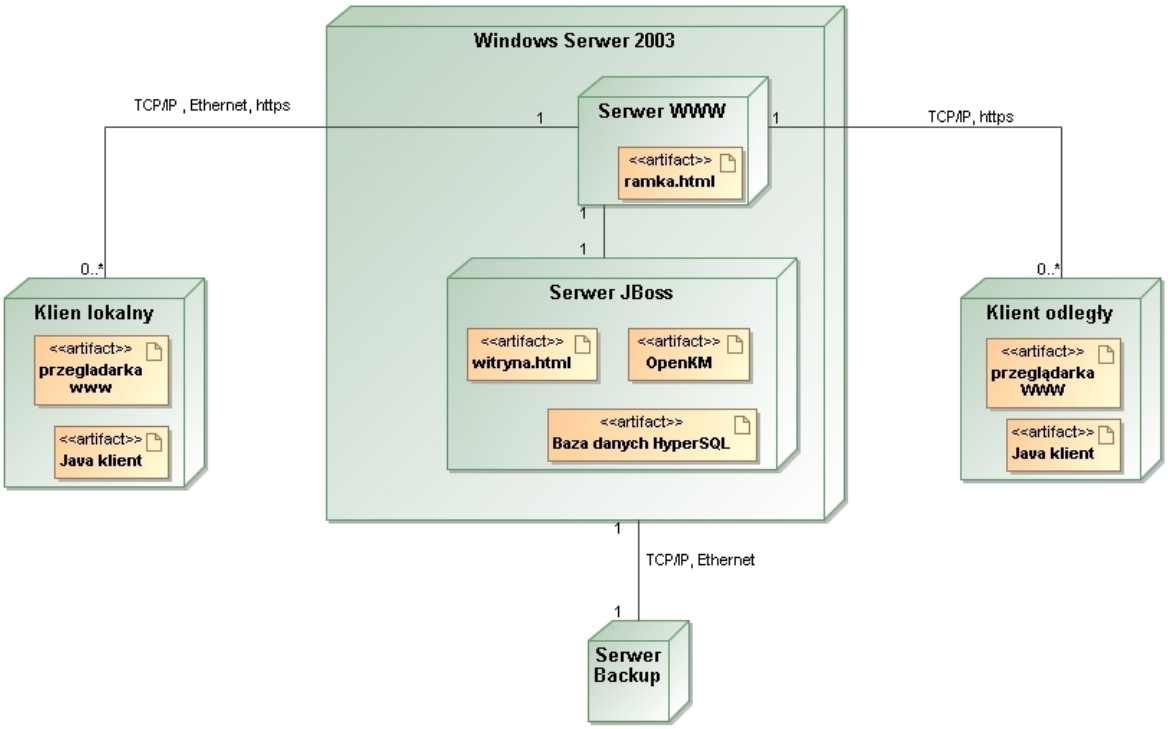
Po tej operacji klienci mają możliwość korzystania z serwera Windows 2003 przy użyciu bezpiecznego połączenia z wykorzystaniem protokołu https.

Konfiguracja bezpiecznego połączenia odnośnie samej aplikacji OpenKM została przedstawiona w punkcie 2.3.2.

2.3. Architektura systemu DMS PIAP

Jak wcześniej przedstawiano, architektura systemu DMS obejmuje serwer JBoss posadowiony na serwerze sprzętowym Windows Server 2003 oraz dołączonych do niego użytkowników lokalnych oraz odległych z wykorzystaniem bezpiecznego połączenia internetowego. Do korzystania z systemu wymagane jest zainstalowanie na komputerze stanowiskowym przeglądarki internetowej oraz klienta Java.

Serwer JBoss jest udostępniany klientowi przez Serwer WWW, na którym posadowiona jest witryna internetowa pełniąca rolę ramki. Ramka ta dostosowuje się w sposób dynamiczny do rozdzielczości ekranu klienta. Ramka wyświetla szyfrowaną zawartość witryny Serwera JBoss.



Rys. 1. Architektura systemu DMS PIAP

2.3.1. Bezpieczny dostęp współużytkowników

Bezpieczny dostęp do danych zapewnia protokół https. Zapewnia on szyfrowanie podczas transferu danych serwer-klient Bezpieczny dostęp współużytkowników do wspólnych zasobów to także odpowiednio przypisane prawa dostępu do poszczególnych katalogów i dokumentów. Z dostępu do posadowionych w serwerze dokumentów mogą korzystać wyłącznie użytkownicy uprawnieni.

2.3.2. Bezpieczeństwo danych

Jednym z elementów bezpieczeństwa danych jest ich systematyczne archiwizowanie (backup). Z istoty funkcjonowania serwera JBoss wynika, że zarówno repozytorium jak i wersje plików są zapisywane w jednym pliku *.dat. Z tego powodu jako metodę archiwizacji przyjęto backup pełny. Backup zachodzi w dwóch fazach: faza pierwsza, to archiwizacja w serwerze Windows 2003, na którym jest zaimplementowany JBoss, faza druga, to przetwarzanie kopii bezpieczeństwa w serwerze backupowym PIAP.

Przed wykonaniem backupu należy zatrzymać server JBoss, uruchamiając przewidziany do tego celu skrypt **shutdown.bat**. Po zatrzymaniu serwera tworzona jest kopia bezpieczeństwa całego katalogu głównego, w którym znajduje się zarówno serwer (pliki konfiguracyjne, baza danych HyperSQL) jak i repozytorium (pliki w formacie *.dat). Po utworzeniu kopii należy uruchomić serwer przy użyciu przewidzianego do tego celu skryptu **run.bat**. W przypadku udostępniania serwera z opcją do korzystania przez użytkowników zewnętrznych skrypt ten uruchamia się z **parametrem –b 0.0.0.0**

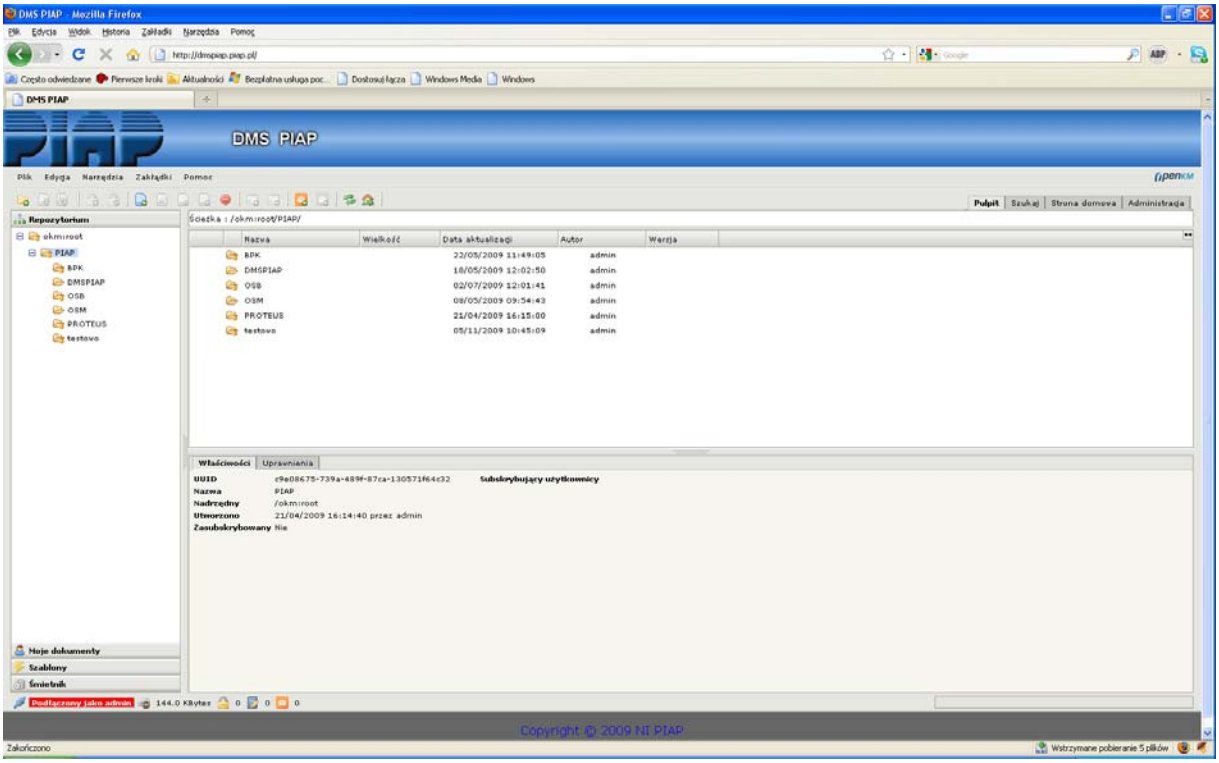
Kopia bezpieczeństwa DMS jest posadawiana w określonej lokalizacji serwera Windows 2003, udostępnionej lokalnemu serwerowi backupowemu PIAP, gdzie zachodzi dalsze przetwarzanie, zgodnie z przyjętą polityką bezpieczeństwa i wg harmonogramu zaprojektowanego w tym serwerze. Przyjęta jest tu zasada, że procesem archiwizacji w fazie drugiej zarządza wyłącznie serwer backupowy.

2.4. Korzystanie z systemu DMSPIAP

2.4.1. Korzystanie z pozycji użytkownika systemu

Jak wynika z funkcjonalności systemu, umożliwia ona współkorzystanie zespołów użytkowników rozproszonych, o odległym dostępie do serwera DMS, z wykorzystaniem narzędzi internetowych. Do zainicjowania pracy w środowisku DMS wymagana jest jedynie przeglądarka internetowa. W celu jak najpełniejszego wykorzystania systemu DMS, zaleca się stosowanie przeglądarki Mozilla Firefox.

Po uruchomieniu przeglądarki i podaniu adresu IP lokalizacji serwera systemu DMS uzyskuje się dostęp do jego zasobów (repozytorium). Sposób prezentacji zasobów, to podzbiór katalogów wchodzących w skład systemu, przypisany do wykorzystywania przez określonego użytkownika. Takie podzbiory są definiowane uprawnieniami dostępu każdego z użytkowników systemu.



Rys. 2. Widok repozytorium systemu DMS PIAP

Zasoby systemu, do których określony użytkownik nie ma uprawnień są dla niego niewidoczne, zasoby z prawem do odczytu oznaczone są kolorem czerwonym, a zasoby z pełnymi prawami są zaznaczone kolorem żółtym. Katalog „Moje Dokumenty" zawiera katalog domowy użytkownika. W systemie DMS definiuje się zarówno użytkowników jak i grupy, do których przypisuje się użytkowników pełniących te same role w sytemie.

2.4.1.1. Wybrane funkcje systemu

e-mail: podczas dodawania dokumentu do określonego katalogu użytkownik ma możliwość poinformowania o tym wskazanych użytkowników systemu. Użytkownicy ci otrzymają wiadomość na swój adres e-mail.

blokada dokumentu: pobranemu z zasobów systemu dokumentowi można przypisać status „zablokowany" na czas przetwarzania go, ograniczając prawo dostępu do niego innym użytkownikom do prawa „tylko do odczytu". Karta Właściwości umożliwia odczyt statusu, w tym informacji o tym, kto aktualnie edytuje ten dokument.

historia: wprowadzenie zmodyfikowanego pliku do systemu powoduje nadanie mu nowego indeksu (numer wersji); poprzednie numery wersji można odczytać na karcie Historia. Można również otworzyć dokument w poprzedniej wersji lub przywrócić dokument zmodyfikowany do wersji poprzedniej wersji z zachowaniem wcześniej przypisanego numeru wersji.

subskrypcja monitorowania dokumentu: po wybraniu takiego żądania, zmiany statusu dokumentu mogą być raportowane użytkownikowi systemu drogą e-mailową.

notatki: każdy plik poddany przetwarzaniu może być opatrzony notatką. Notatka nie podlega modyfikacji przez kolejnych użytkowników. Każda wersja dokumentu może posiadać jedną notatkę.

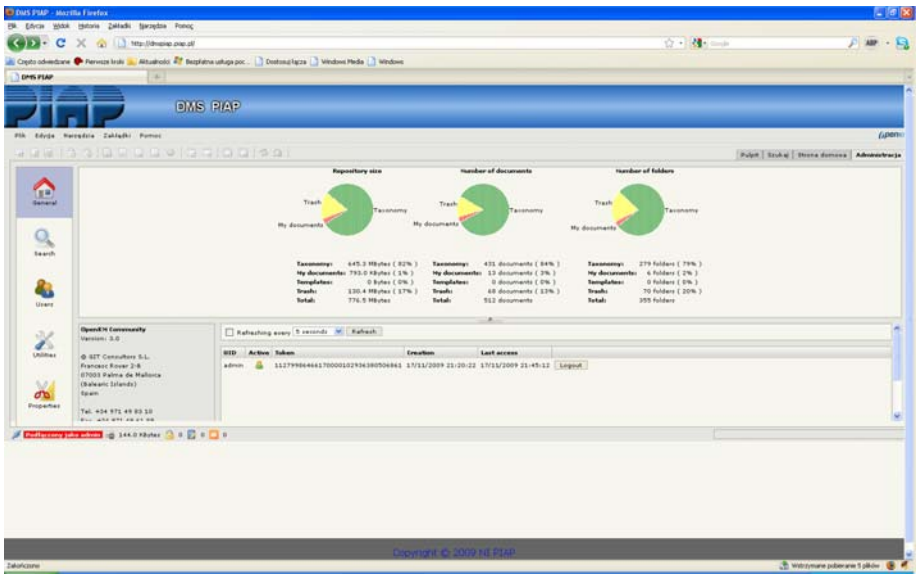
kopiowanie odnośników do dokumentów: ta opcja – dostępna jedynie w odpowiednio skonfigurowanej przeglądarce Mozilla Firefox – umożliwia kopiowanie lokalizacji URL dokumentu z poziomu DMS.

wyszukiwanie: w systemie OpenKM istnieje możliwość pełnotekstowego wyszukiwania danych w wielu powszechnie stosowanych formatach, w tym dokumentach PDF, TXT, MS Office, OpenOffice, a także w znacznikach ID3 dla formatu MP3, czy w znacznikach EXIF (EXtended Interchange Format) dla formatu JPEG.

2.4.2. Korzystanie z pozycji administratora

Administrator ma dostęp do pełnego repozytorium niezależnie od uprawnień nadawanych przez użytkowników w ramach przypisanych im uprawnień.

Administrator swoje zadania wykonuje na karcie Administracja



Rys. 3. Widok statystyk systemu DMS PIAP

2.4.2.1. Wybrane funkcje Administratora systemu

Przeglądanie statystyk: Administrator ma dostęp do statystyk DMS takich jak rozmiar repozytorium, liczba dokumentów oraz liczba folderów

Zarządzanie sesjami użytkowników: Administrator monitoruje bieżące sesje użytkowników w systemie. Ma on przy tym prawo do przerwania sesji użytkownika (zabicie sesji), co jest istotne zwłaszcza przy usuwaniu zawieszonych na okres karencji sesji (zombie) pozostałych po niezamierzonym przez użytkownika zerwaniu komunikacji klient-serwer.

Nadawanie uprawnień do katalogów i dokumentów: Administrator nadaje uprawnienia do katalogów lub dokumentów użytkownikom i grupom, zgodnie z polityką właściciela DMS (Kierownika projektu PROTEUS). Opcje uprawnień to: odczyt i/lub zapis. Użytkownik posiadający prawo odczytu i zapisu do określonego podzestawu repozytorium, może nadawać prawa innym użytkownikom i grupom w nadzorowanym przez siebie obrębie. Nadawanie uprawnień do poszczególnych katalogów lub plików przeprowadza się w repozytorium, poprzez wybranie określonego zasobu i przypisanie mu wybranych uprawnień.

Zarządzanie użytkownikami i grupami użytkowników: zgodnie z polityką właściciela DMS, Administrator wprowadza lub usuwa użytkowników. Dodając użytkownika określa się jego UID (Unikalny Identyfikator w formacie ciągu znakowego), wprowadza się adres e-mail stosowany przez tego użytkownika, jego hasło do systemu, określa się status aktywności (aktywny-nieaktywny) oraz przypisuje mu się rolę (grupę). Każdy użytkownik musi mieć przypisaną co najmniej rolę User Role, która uprawnia do używania systemu. Liczba ról nie jest ograniczona. Użytkownicy pełniący te same role łączeni są w grupy. Ułatwia to zarządzanie uprawnieniami użytkowników, realizowanymi w tej opcji zbiorczo. W systemie DMS nazwy grup są tożsame z nazwami ról.

Przeglądanie zdarzeń: ta opcja umożliwia monitorowanie operacji wykonywanych w systemie przez dowolnego użytkownika lub przez system.

Wyszukiwanie: poza możliwościami wyszukiwania dostępnymi dla użytkownika, administrator dysponuje rozszerzonymi opcjami takimi jak wyszukiwanie zablokowanych dokumentów oraz pisanie zapytań SQL do bazy.

3. PODSUMOWANIE

Systemy DMS są w Polsce wykorzystywane w małym stopniu – świadczy o tym liczba dostępnych publikacji na ten temat. Spowodowane jest to niewiedzą potencjalnych odbiorców. Taki system można zastosować w każdej firmie, która ma z dużymi ilościami dokumentów tworzonych przez różne osoby.
Warto zainteresować się tego typu rozwiązaniami, aby podnieść wydajność pracy w swojej firmie.

4. ZAŁĄCZNIK A

Konfiguracja pliku

C:\WWW\dmspiap\ server\default\deploy\jboss-web.deployer\serwer.xml

```
<Server>
 <Listener className="org.apache.catalina.core.AprLifecycleListener" SSLEngine="on" />
 <Listener className="org.apache.catalina.core.JasperListener" />
  <Service name="jboss.web">
   <Connector port="8088" address="${jboss.bind.address}"
      maxThreads="250" maxHttpHeaderSize="8192"
      emptySessionPath="true" protocol="HTTP/1.1"
      enableLookups="false" redirectPort="7443" acceptCount="100"
      connectionTimeout="20000" disableUploadTimeout="true" />
         <Connector port="7443" protocol="HTTP/1.1" SSLEnabled="true"
          maxThreads="150" scheme="https" secure="true"
           clientAuth="false"
    strategy="ms"
          address="${jboss.bind.address}"
          keystoreFile="${jboss.server.home.dir}/conf/pryszawa"
          keystorePass="password"
          truststoreFile="${jboss.server.home.dir}/conf/pryszawa"
          truststorePass="password"
          sslProtocol="TLS"/>
     <Engine name="jboss.web" defaultHost="localhost">
       <Realm className="org.jboss.web.tomcat.security.JBossSecurityMgrRealm"
         certificatePrincipal="org.jboss.security.auth.certs.SubjectDNMapping"
         allRolesMode="authOnly"/>
       <Host name="localhost"
         autoDeploy="false" deployOnStartup="false" deployXML="false"
         configClass="org.jboss.web.tomcat.security.config.JBossContextConfig" >
          <Valve className="org.apache.catalina.authenticator.SingleSignOn" />
          <Valve className="org.jboss.web.tomcat.service.jca.CachedConnectionValve"

cachedConnectionManagerObjectName="jboss.jca:service=CachedConnectionManager"
             transactionManagerObjectName="jboss:service=TransactionManager" />
       </Host>
     </Engine>
   </Service>
</Server>
```

5. ZAŁĄCZNIK B

Konfiguracja pliku

C:\WWW\dmspiap\ server\default\deploy\mail-service.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- $Id: mail-service.xml 62349 2007-04-15 16:48:15Z dimitris@jboss.org $ -->
<server>
 <!-- Mail Connection Factory                                    -->
 <mbean code="org.jboss.mail.MailService"
      name="jboss:service=OpenKM">
  <attribute name="JNDIName">java:/mail/OpenKM</attribute>
  <attribute name="User">user</attribute>
  <attribute name="Password">password</attribute>
  <attribute name="Configuration">
   <!-- A test configuration -->
   <configuration>
                <!-- Change to your mail server prototocol -->
    <property name="mail.store.protocol" value="pop3"/>
    <property name="mail.transport.protocol" value="smtp"/>
    <!-- Change to the user who will receive mail  -->
    <property name="mail.user" value="user"/>
    <!-- Change to the mail server  -->
    <property name="mail.pop3.host" value="ni.piap.pl"/>
    <!-- Change to the SMTP gateway server -->
    <property name="mail.smtp.host" value="ni.piap.pl"/>
    <!-- The mail server port -->
    <property name="mail.smtp.port" value="25"/>
                <property name="mail.smtp.starttls.enable" value="true"/>
                <property name="mail.smtp.auth" value="true"/>
    <!-- Change to the address mail will be from  -->
    <property name="mail.from" value="dmspiap@piap.pl"/>
    <!-- Enable debugging output from the javamail classes -->
    <property name="mail.debug" value="false"/>
   </configuration>
  </attribute>
  <depends>jboss:service=Naming</depends>
 </mbean>
</server>
```

6. ZAŁĄCZNIK C

Konfiguracja pliku

C:\WWW\dmspiap\OpenKM.cfg

```
# Default configuration values
# Default configuration values
# repository.config=repository.xml
# repository.home=repository
# system.user=system
# default.user.role=UserRol
# default.admin.role=AdminRol
# principal.adapter=es.git.openkm.core.principal.DatabasePrincipalAdapter
 max.file.size=100
# max.search.results=25
# system.demo=off
# system.ocr=/usr/local/bin/tesseract
application.url=https://dmspiap.piap.pl:7443/OpenKM/es.git.openkm.frontend.Main/index.jsp
# default.lang=
subscription.message.subject=DMS PIAP - subskrypcja {2}
subscription.message.body=<b>Dokument: </b>{2}<br/><b>Link: </b>{0}<br/><b>Uzytkownik:
</b>{3}<br/><b>Zdarzenie: </b>{4}<br/><b>Komentarz:</b>{5}
notify.message.subject=DMS PIAP - Powiadomienie
notify.message.body=<b>Uzytkownik: </b>{3}<br/><b>Nazwa Dokumentu: </b>{2}<br/><b>Link:
</b>{0}<br/><b>Sciezka: </b>{1}<br/><b>Wiadomosc: </b>{4}<br/>,
```

gdzie:

Cyfry od 0 do 5 są to zmienne, które wykorzystujemy do spersonalizowania otrzymywanych od systemu wiadomości email

	{5} - comment
SUBSCRIPTION SUBJECT	
{0} - event type	NOTIFY SUBJECT
{1} - document path	{0} - document path
{2} - document name	{2} - document name
SUBSCRIPTION BODY	NOTIFY BODY
{0} - document url	{0} - document url
{1} - document path	{1} - document path
{2} - document name	{2} - document name
{3} - user id	{3} - user id
{4} - event type	{4} - message

7. LITERATURA

1. Praca Zbiorowa, 2005 r., JBoss 4.0 Podręcznik Administratora, Helion
2. Praca zbiorowa, 2005 r., J2EE. Vademecum profesjonalisty, Helion
3. Bill Burke, Richard Monson-Haefel , 2007 r., Enterprise JavaBeans 3.0, Helion
4. M. Yuan, 2007 r., JBoss Seam, Prentice Hall