Ω-Ψ^{R}

Uczelniana Baza Wiedzy

Instalacja i konfiguracja systemu

Wersja 1.2.2

Wydział Elektroniki i Technik Informacyjnych

Politechnika Warszawska

To opracowanie jest zrealizowane w ramach programu strategicznego: Interdyscyplinarny system interaktywnej informacji naukowej i naukowo technicznej, finansowanego przez Narodowe Centrum Badań i Rozwoju nr umowy SP/I/1/77065/10

This work is supported by the National Centre for Research and Development (NCBiR) under Grant No. SP/I/1/77065/10 by the Strategic scientific research and experimental development program: „Interdisciplinary System for Interactive Scientific and Scientific-Technical Information”.

Historia dokumentu

Numer wersji	Data wersji	Opis zmian	Autor

Spis treści

0 Używane skróty

API	Application Programming Interface
BG PW	Biblioteka Główna Politechniki Warszawskiej
JCR	Java Content Repository
SQL	Structured Query Language
WEiTI	Wydział Elektroniki i Technik Informacyjnych
WUT	Warsaw University of Technology
WWW	World Wide Web
XML	eXtensible Markup Language
XSD	XML Schema Definition

1 Wstęp

Pod kierunkiem prof. Henryka Rybińskiego, w Instytucie Informatyki Politechniki Warszawskiej został opracowany system komputerowy do utrzymywania uczelnianego repozytorium dorobku publikacyjnego pracowników uczelni. Repozytorium zostało utworzone z wykorzystaniem oprogramowania OMEGA-PSIR (Ω-Ψ^R):

http://omegapsir.ii.pw.edu.pl

System ten zaimplementowano z wykorzystaniem narzędzi programistycznych technologii języka Java – JCR (*Java Content Repository*). Jako bazę danych wykorzystuje się bazy systemu MySQL.

Wszyscy użytkownicy systemu (użytkownicy końcowi, redakorzy, administratorzy) posiadają dostęp do właściwych dla nich funkcji systemu z wykorzystaniem przeglądarki internetowej.

Pierwszego wdrożenia dokonano na Wydziale Elektroniki i Technik Informacyjnych Politechniki Warszawskiej.

Obecnie system jest wdrażany na Politechnice Warszawskiej i obejmuje wszystkie wydziały uczelni.

Niniejszy podręcznik opracowano na potrzeby administratorów, w celu ułatwienia procesu instalacji i konfiguracji systemu.

1. Opis ogólny

Repozytorium Uczelniane wykorzystywane jest to gromadzenia opisów podstawowych efektów działalności naukowo-badawczej pracowników Uczelni i jest rdzeniem Uczelnianej Bazy Wiedzy.

System Repozytorium został zaimplementowany z wykorzystaniem narzędzi programistycznych technologii języka Java – JCR (*Java Content Repository*). Jako bazę danych wykorzystuje się bazy systemu MySQL.

Wersja produkcyjna Bazy Wiedzy PW jest dostępne pod adresem:

http://repo.bg.pw.edu.pl

Oprogramowanie zostało zaimplementowane z wykorzystaniem platformy Java EE oraz istniejących bibliotek wolnego oprogramowania.

a) Szkielet aplikacji – użyty został szkielet Seam (http://seamframework.org/), ułatwiający tworzenie ergonomicznych i łatwo integrowanych aplikacji
b) Dane – system opiera się na wykorzystaniu uznanego formatu XML. Struktura danych weryfikowana jest poprzez definicje XML Schema Definition (XSD). Dane następnie konwertowane są do modelu obiektowego za pomocą technologii JAXB (http://jaxb.java.net/). Dane utrwalane są w systemie repozytoryjnym Jackrabbit (http://jackrabbit.apache.org/) zgodnym ze standardem Java Content Repository (JSR 170 i JSR 283). Standard JCR umożliwia między innymi: indeksowanie, wersjonowanie, kontrole uprawnień, łatwą integrację na poziomie danych, obserwacje zmian w treści, swobodną zmianę dostawcy implementacji.
   Do celów technicznej konfiguracji system korzysta również z relacyjnej bazy danych celem przechowywania adresów do zdalnych repozytoriów oraz danych niezbędnych do autoryzacji użytkowników
c) Warstwa prezentacji – oparta na technologii Java Server Faces z użyciem biblioteki RichFaces (http://www.jboss.org/richfaces) – opartej na podejściu AJAX. Bazowy interfejs użytkownika generowany jest automatycznie na podstawie definicji XSD a

następnie może być swobodnie dostosowywany do specyficznych potrzeb poszczególnych funkcjonalności, co zostało pokazane w dalszej części dokumentacji.

d) Warstwa logiki biznesowej – zawarta jest w komponentach EJB 3.0 (http://jcp.org/aboutJava/communityprocess/final/jsr220/), co zapewnia m.in. zachowanie transakcyjne i łatwość ekspozycji do systemów zewnętrznych.

Wdrożony system udostępnia następujące funkcje:

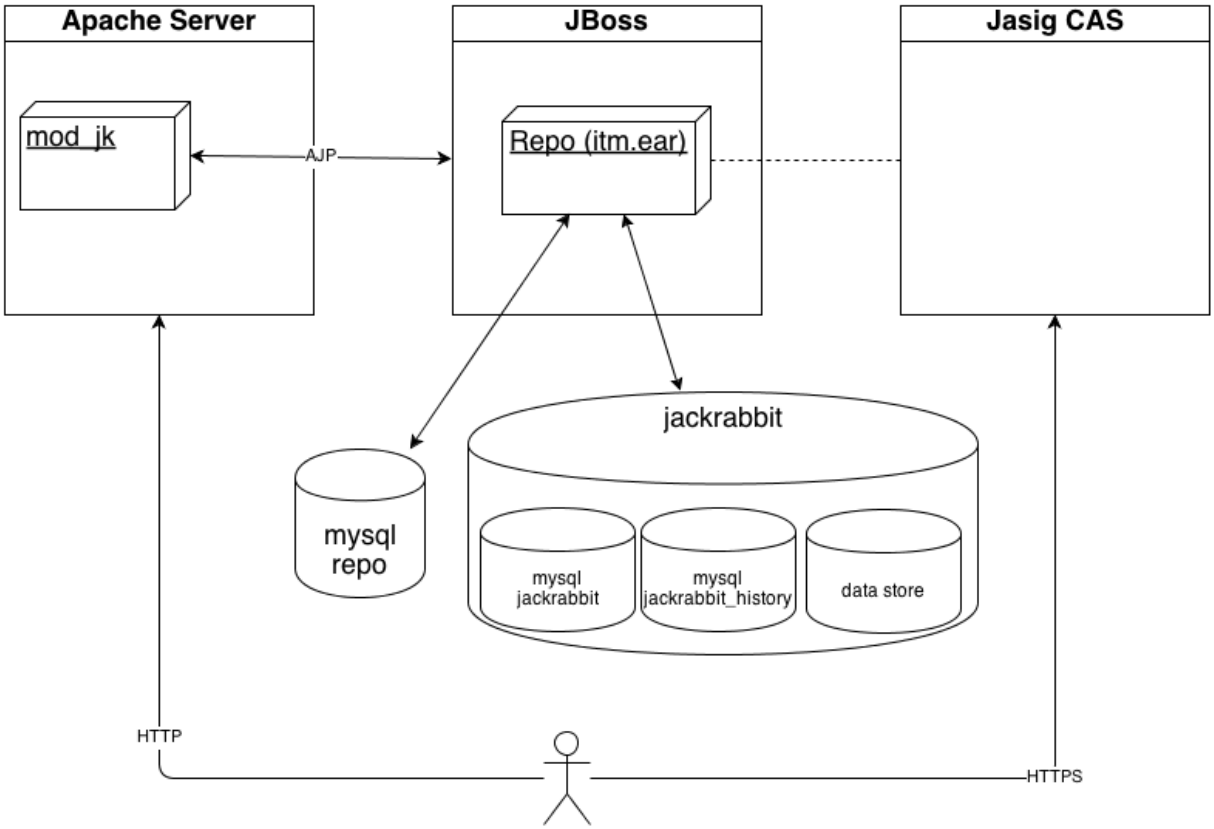
- przechowuje i udostępnia zasoby bibliograficzne min
  - publikacje (książki, artykuły, rozdziały z książek, materiały konferencyjne i inne)
  - dyplomy (inżynierskie, magisterskie, doktorskie)
  - projekty realizowane przez uczelnię
  - patenty
  - inne dokumenty (raporty etc.)
  - autorów wraz z afiliacją
  - aktywność naukowców

- przechowuje i udostępnia dane dotyczące pracowników oraz ich dorobku
- zapewnia możliwości oceny dorobku pracowników i jednostek organizacyjnych
- udostępnia raporty na cele wewnętrzne i ministerialne
- zapewnia import/export danych a także integracje z innymi systemami wydziałowymi
- umożliwia łatwą rozbudowę

2. Oprogramowanie i bazy danych

2.1. Ogólna architektura systemu

Poniższy rysunek przedstawia ogólną architekturę systemu i zasady współdziałania jego podsystemów.



2.2. Oprogramowanie

System działa pod kontrolą wirtualnej maszyny Javy w wersji 1.6 oraz serwera aplikacji JBoss w wersji 4.2.2.

2.3. Bazy danych

System został skonfigurowany do działania z bazą mySQL, ale możliwe jest wykorzystanie prawie dowolnej bazy danych.

W mySQL zostały utworzone trzy bazy danych:

- repo – zawiera informacje o administratorach i przykładowych rolach redaktorów repozytorium. Kolejnych użytkowników można już stworzyć w systemie.
- jackrabbit – zawiera dane bibliograficzne
- jackrabbit-history – zawiera historie modyfikacji danych bibliograficznych

Bazy jackrabbit są tworzone i utrzymywane przez system.

Schemat bazy repo jest tworzony przez system, ale wymaga ręcznego wypełnienia przy pomocy narzędzi mysql.

Dołączane pliki binarne przechowywane są na dysku w dedykowanym katalogu zasobów repozytorium.

2.4. System CAS

Dodatkowo, w celu ułatwienia uwierzytelniania redaktorów i użytkowników do zasobów chronionych wykorzystywany jest system centralnego Uwierzytelniania użytkowników CAS.

System ten został wdrożony na bazie oprogramowania utworzonego na Yale University z wykorzystaniem technologii programowania w języku Java, zaś od 2004 roku rozwijanego w ramach projektu Jasig (http://www.jasig.org/cas/). System CAS działa pod nadzorem serwera Tomcat.

Centralny System Uwierzytelniania (*ang*. Central Authentication Service - CAS) zapewnia realizację zasady pojedynczego logowania (*ang*. Single Sign On) w środowisku, gdzie użytkownik korzysta z wielu różnorodnych aplikacji z chronionym dostępem. W przypadku wejścia do chronionej strefy jednej z aplikacji użytkownik musi załogować się, aby przejść fazę uwierzytelniania. Jednak od tego momentu, wejście do innych aplikacji korzystających z

systemu CAS nie wymaga ponownego logowania. Usprawnia to w znacznym stopniu pracę użytkowników z różnymi aplikacjami. Dzięki temu, można stworzyć system składający się z wielu aplikacji, które przy niewielkich zabiegach mogą stwarzać wrażenie jednej spójnej aplikacji, do której użytkownik loguje się tylko raz.

Obecnie wykorzystywana wersja systemu CAS zapewnia logowanie użytkowników WEiTI oraz edytorów z całej uczelni.

3. Wymagania sprzętowe

1. Procesor: 4 - 8 rdzeniowy
2. Pamięć RAM: 8 -16 GB (ze wskazaniem na 16GB)
3. Pamięć dyskowa:
    a. System - 70 GB
    b. Pamięć dyskowa do przechowywania metadanych i dokumentów źródłowych, przy założeniu 2000 publikacji i 5000 dyplomów na rok:
        - 1MB * 2000publikacji + 6MB * 5000dyplomów = 32GB /1 rok
        - zwiększone podwójnie na indeks daje około 64GB/rok

    Roczny przyrost danych wraz z indeksem można szacować na około 100GB/ rok

4. Wymagania dotyczące oprogramowania

System Repozytorium wymaga następujących narzędzi oprogramowania:

- System operacyjny Linux (sprawdzono Linux Debian oraz Linux RedHat) lub Windows
- Serwer aplikacyjny JBoss wersji 4.2.2.GA (nie należy stosować wersji wyższych niż 4.2.3)
    - Serwer JBoss wymaga oprogramowania Java JDK wersji co najmniej 1.6
- Relacyjna baza danych - dotychczas wykorzystywano MySQL wersji 5.1

- System uwierzytelniania CAS (Central Authentication System) – Jasig CAS

Ze względów bezpieczeństwa, do udostępnienia aplikacji w środowisku Web, zalecane jest wykorzystanie serwera WWW – Apache, który stanowi serwer proxy i pośredniczy pomiędzy serwerem aplikacji JBoss a użytkownikami sieci Web.

5. Przygotowanie środowiska systemowego

Wymagane kroki instalacyjne zostaną opisane na przykładzie instalacji systemu w środowisku Linux Debian

1. Należy zapewnić prawidłowe działanie wymaganych narzędzi systemowych;

    - oprogramowanie serwera Apache
    - oprogramowanie MySQL

2. Instalacja oprogramowania Java JDK

3. Instalacja (rozpakowanie) oprogramowania serwera JBoss, np. w katalogu /var/lib

4. Zalecane jest utworzenie dedykowanego konta użytkownika – właściciela aplikacji systemu Repozytorium (np. repo) i przypisanie temu użytkownikowi praw właściciela wszystkim plikom zainstalowanego oprogramowania JBoss

5. Zalecane jest utworzenie dedykowanego użytkownika mySQL (np. repo)

6. Sprawdzenie, czy serwer JBoss działa prawidłowo

    - Należy uruchomić serwer JBoss, np. uruchamiając go lokalnie za pomocą polecenia:
      /var/lib/jboss-4.2.2.GA/bin/run.sh –b localhost

- Korzystając z lokalnego klienta webowego można sprawdzić, czy serwer JBoss odpowiada, np.

  links http://localhost:8080

6. Instalacja aplikacji systemu Repozytorium

1. Należy utworzyć katalog aplikacji przeznaczony do przechowywania danych (część danych JCR umieszcza w bazie mySQL), indeksów, baz pomocniczych oraz plików kontrolnych i narzędzi użytkowych aplikacji systemu Repozytorium. Z tego względu katalog ten powinien być umieszczony na dysku o odpowiedniej pojemności. Przykładowo, może to być katalog /data

2. W katalogu aplikacji (np. /data) należy rozpakować katalog repo znajdujący się w dostarczonej paczce instalacyjnej systemu Repozytorium, co powinno spowodować utworzenie następującej struktury:

```
/data/repo
/data/repo/cacerts
/data/repo/jackrabbit
/data/repo/jackrabbit_backup
/data/repo/jsf-libs
/data/repo/ph
/data/repo/repo-init
/data/repo/tools
```

3. Należy zapewnić prawidłową łączność serwera JBoss w trybie SSL z serwerem CAS, w tym celu można posłużyć się odpowiednio przygotowanym plikiem certyfikatów (np. cacerts.jks), który można umieścić np. w katalogu /data/repo/cacerts (odwołanie do tej lokalizacji podaje się w pliku run.conf

4. Należy zastosować następujące ustawienia konfiguracyjne serwera JBoss:
   - Zwiększenie w definicji zmiennej JAVA_OPTS (zwykle w pliku /var/lib/jboss-4.2.2.GA/bin/run.conf) używanej pamięci RAM oraz zapewnienie prawidłowej łączności serwera JBoss w trybie SSL z serwerem CAS (-server, -Xms, -Xmx, -XX:MaxPermSize, -Djavax.net.ssl.trustStore), a także ułatwienie diagnostyki, przykładowo:

```
#if [ "x$JAVA_OPTS" = "x" ]; then
   JAVA_OPTS="-server -Xms4096m -Xmx4096m -XX:MaxPermSize=512m
       -Dsun.rmi.dgc.client.gcInterval=3600000
       -Dsun.rmi.dgc.server.gcInterval=3600000
       -Djavax.net.ssl.trustStore=/data/repo/cacerts/cacerts.jks
```

```
        -Dcom.sun.management.jmxremote
        -XX:+HeapDumpOnOutOfMemoryError
        -XX:HeapDumpPath=/home/repo/heap"
#fi
```

- Dodanie w definicji zmiennej JAVA_OPTS (zwykle w pliku /var/lib/jboss-4.2.2.GA/bin/run.conf) specyfikacji zapewniającej prawidłowe wyświetlanie polskich znaków diakrytycznych:
  -Dfile.encoding=utf-8

- Dodatkowa uwaga dotyczy instalacji działającej z wykorzystaniem oprogramowania Java JDK wersji 1.8 (lub wyższych). W takim przypadku w powyższej specyfikacji trzeba dodać dodatkowy parametr:

  -Ddrools.dialect.java.compiler=JANINO

5. W katalogu aplikacji serwera JBoss (np. /var/lib/jboss-4.2.2.GA/server/default/deploy) należy rozpakować zawartość katalogu repo-jboss-deploy znajdujący się w dostarczonej paczce instalacyjnej systemu Repozytorium:

   - kopiowane pliki obejmują:
     - itm-ear-x.y.z.ear - plik aplikacji (np. itm-ear-1.1.9.ear)
     - jackrabbit-jca-2.8.0.rar - jackrabbit connector
     - jcr-ds.xml - konfiguracja dostępu jackrabbit
     - RepoPW-prod-ds.xml - konfiguracja dostępu do bazy mySQL
     - repo-service.xml - konfiguracja aplikacji Repozytorium
     - oaicat.war - plik aplikacji OAI-PMH (dotyczy tylko PW)

   - trzeba zmodyfikować pliki konfiguracyjne xml, aby zawierały specyfikacje odpowiadające aktualnej konfiguracji, przykładowe ustawienia w tych plikach:

     - jcr-ds.xml

```
<connection-factories>.
  <tx-connection-factory>
      <jndi-name>jcr/repository</jndi-name>.
      <rar-name>jackrabbit-jca-2.8.0.rar</rar-name>
      <connection-definition>javax.jcr.Repository</connection-definition>
      <config-property name="configFile"
      type="java.lang.String">/data/repo/jackrabbit/repository.xml</config-
      property>
      <config-property name="homeDir"
      type="java.lang.String">/data/repo/jackrabbit</config-property>
```

```
        <max-pool-size>230</max-pool-size><!-- adjusted in order to avoid JCA
        shortage from Jackrabbit JCA -->.
        <config-property name="bindSessionToTransaction"
        type="java.lang.Boolean">true</config-property>
        <xa-transaction/>
    </tx-connection-factory>
</connection-factories>
```

- RepoPW-prod-ds.xml

```
<datasources>
   <local-tx-datasource>
      <jndi-name>RepoPWDatasource</jndi-name>
      <use-java-context>false</use-java-context>
      <connection-url>jdbc:mysql://localhost:3306/repo</connection-url>
      <driver-class>com.mysql.jdbc.Driver</driver-class>
      <user-name>repo</user-name>
      <password>xxxxxxxxxx</password>
   </local-tx-datasource>
</datasources>
```

- konfiguracja dostępu do bazy mySQL w pliku RepoPW-prod-ds.xml wymaga zainstalowania odpowiedniego sterownika JDBC w katalogu serwera JBoss server/default/lib. W przypadku bazy mySQL można skopiować plik sterownika z katalogu repo/tools/mysql-connector-java-x.x.xx-bin.jar

6. Zalecane jest zapewnienie automatycznego startowania serwera JBoss w przypadku restartu systemu operacyjnego (zwykle używa się do tego celu standardowego skryptu z opcjami start|stop|restart)

7. Adaptacja skryptów tworzenia i odtwarzania kopii bezpieczeństwa.
   W katalogu aplikacji repo/tools znajdują się skrypty wykorzystywane do tworzenia i odtwarzania kopii bezpieczeństwa:

backup tworzenie kopii bezpieczeństwa bazy jackrabbit, która powstaje jako kopia bazy jackrabbit w katalogu jackrabbit_backup

restore_from_backup odtworzenie kopii bezpieczeństwa bazy jackrabbit z kopii bezpieczeństwa przechowywanej w katalogu jackrabbit_backup

W skryptach tych należy dokonać modyfikacji, aby używane ścieżki odzwierciedlały aktualną konfigurację systemu

W niektórych instalacjach stwierdzono, że standardowa konfiguracja systemu MySQL nie określa parametrów umożliwiających prawidłowe działanie procedury odtwarzania kopii bezpieczeństwa. W trakcie działania procedury może wówczas pojawić się diagnostyka zawierająca wskazanie nieprawidłowej konfiguracji mySQL, np.:

```
com.mysql.jdbc.PacketTooBigException: Packet for query is too large
(2019812 > 1048576). You can change this value on the server by
setting the max_allowed_packet' variable.
         at com.mysql.jdbc.MysqlIO.send(MysqlIO.java:3248)
         at com.mysql.jdbc.MysqlIO.sendCommand(MysqlIO.java:1940)
         at com.mysql.jdbc.MysqlIO.sqlQueryDirect(MysqlIO.java:2113)
         at com.mysql.jdbc.ConnectionImpl.execSQL(ConnectionImpl.java:2568)
         at
com.mysql.jdbc.PreparedStatement.executeInternal(PreparedStatement.jav
a:2113)
```

Należy odpowiednio skonfigurować system mySQL, w tym przypadku należy w pliku my.conf ustawić paramter max_allowed_packet, np.

```
max_allowed_packet = 50M
```

7. Instalacja baz danych

1. Należy utworzyć 3 bazy danych mySQL i zapewnić dostęp do tych baz użytkownikowi repo:
   repo
   jackrabbit
   jackrabbit_history
2. Baza repo powinna być zainicjowana z wykorzystaniem dostarczonego pliku repo-db.sql zawierającego odpowiednie polecenia SQL. Plik ten znajduje się w katalogu aplikacji repo/repo-init.

W przypadku, gdy instalacja nie dotyczy PW, po rozpakowaniu pliku należy zmienić w nim oznaczenie skrótowe przyjęte dla afiliacji o identyfikatorze id=10, które obecnie jest ustawione na ‘PW’, powinno być zgodne z polskim skrótem przyjętym dla uczelni. Po dokonaniu tej modyfikacji można zastosować polecenie:

```
mysql -p -u repo repo < repo-db-init.sql
```

Jeśli dane zostały załadowane bez wykonanie modyfikacji afiliacji o identyfikatorze id=10. Można dokonać odpowiedniej zmiany wykorzystując polecenie SQL:

```
UPDATE Affiliation SET name='XX' WHERE id=10;
```

gdzie XX należy zastąpić polskim skrótem przyjętym dla uczelni.

W utworzonym podkatalogu aplikacji repo/jackrabbit należy zmodyfikować plik konfiguracyjny repository.xml, aby zawierał poprawne parametry dostępu do bazy jackrabbit w mySQL. Parametry te występują w dwóch definicjach elementu <PersistenceManager>. Przykładowo:

```
        <PersistenceManager
class="org.apache.jackrabbit.core.persistence.pool.MySqlPersistenceManager">
          <param name="url" value="jdbc:mysql://localhost/jackrabbit"/>
             <param name="user" value="repo"/>
             <param name="password" value="xxxxxxxxxx"/>
          <param name="schemaObjectPrefix" value="${wsp.name}_"/>
          <param name="bundleCacheSize" value="80"/>
    </PersistenceManager>

        <PersistenceManager
class="org.apache.jackrabbit.core.persistence.pool.MySqlPersistenceManager">
          <param name="url" value="jdbc:mysql://localhost/jackrabbit_history"/>
             <param name="user" value="repo"/>
```

```
        <param name="password" value="xxxxxxxxxx"/>
      <param name="schemaObjectPrefix" value="version_"/>
    </PersistenceManager>
```

3. Odtworzenie startowej bazy Repozytorium z kopii bezpieczeństwa.
   Wykonanie tego kroku jest opcjonalne i zależy od tego, czy tworzymy instalację z pustą bazą danych, czy też z zainicjowaną bazą danych dostarczoną przez dystrybutora. Dostarczona w paczce instalacyjnej baza danych (jako kopia bezpieczeństwa) zawiera dane systemowe i walidacyjne opisane w p. 10 i jej odtworzenie nie będzie wymagało wykonywania operacji opisanych w p. 10.

   Należy sprawdzić, czy skrypt odtwarzania został odpowiednio przystosowany do aktualnej konfiguracji. Jako użytkownik ‘repo’, przechodzimy do katalogu aplikacji repo/tools i wykonujemy polecenie:

   ```
   ./restore_from_backup
   ```

   Wykonywana jest czasochłonna operacja odtwarzania bazy jackrabbit.

8. Konfiguracja aplikacji systemu Repozytorium

1. Konfiguracja dostępu aplikacji do serwera CAS
   Domyślnie aplikacja korzysta z serwera CAS dostępnego pod adresem:
   https://saturn.elka.pw.edu.pl/cas

   Specyfikacje dotyczące komunikacji z serwerem CAS znajdują się pliku konfiguracji aplikacji umieszczonym jako
   /var/lib/jboss-4.2.2.GA/server/default/deploy/repo-service.xml.

   Zmiany dotyczące dostępu do serwera CAS trzeba dokonać w dwóch parametrach (zaznaczonych poniżej kolorem zielonym), natomiast określnie adresu serwera powrotu (adres instalacji repozytorium) wykonuje się w jednym parametrze (zaznaczonych poniżej kolorem żółtym).
   Jeśli zachowujemy istniejący adres serwera CAS, wystarczy dokonać zmiany adresu powrotu, aby zapewnić, że po zalogowaniu użytkownik będzie przenoszony do właściwej instalacji repozytorium. Przykładowe specyfikacje pokazano poniżej:

```
<jndi:binding name="omega-psir/cas/casServerLoginUrl">
    <jndi:value trim="true">https://saturn.elka.pw.edu.pl/cas/login</jndi:value>
</jndi:binding>
<jndi:binding name="omega-psir/cas/casServerUrlPrefix">
    <jndi:value trim="true">https://saturn.elka.pw.edu.pl/cas</jndi:value>
</jndi:binding>
<jndi:binding name="omega-psir/cas/serverName">
    <jndi:value trim="true">http://repo.pw.edu.pl</jndi:value>
</jndi:binding>
```

Po skonfigurowaniu dostępu do serwera CAS, warto sprawdzić, czy działa prawidłowo logowanie i korzystanie z aplikacji Repozytorium w trybie administratora/redaktora:

- Należy na nowo uruchomić serwer JBoss, aby zostały zaakceptowane wprowadzone zmiany
- Korzystając z dowolnej przeglądarki sprawdzamy, czy serwer JBoss udostępnia aplikację Repozytorium do logowania. Dostarczona konfiguracja umożliwia dostęp administratorowi Repozytorium, któremu nadano identyfikator testadm. Hasło będzie przekazane oddzielnie.

2. Konfiguracja głównego identyfikatora struktury uczelni
Aplikacja wykorzystuje parametr instanceCode zdefiniowany w pliku konfiguracyjnym repo-service.xml do ustalenia początku hierarchicznej struktury jednostek uczelni. Parametr ten powinien być zgodny ze polskojęzycznym skrótem nazwy uczelni w rekordzie afiliacji dotyczącym uczelni (Akronim w języku polskim).
Specyfikację dla Politechniki Warszawskiej pokazano poniżej:

```
<jndi:binding name="omega-psir/instanceCode">
    <jndi:value trim="true">PW</jndi:value>
</jndi:binding>
```

gdzie PW jest skrótem uczelni określonym w rekordzie Politechniki Warszawskiej.

3. Konfiguracja przedrostka identyfikatorów rekordu
Aplikacja wykorzystuje parametr idPrefix zdefiniowany w pliku konfiguracyjnym repo-service.xml do ustalenia początkowego elementu identyfikatorów tworzonych rekordów.
Specyfikację dla Politechniki Warszawskiej pokazano poniżej:

```
<jndi:binding name="omega-psir/idPrefix">
    <jndi:value trim="true">WUT</jndi:value>
</jndi:binding>
```

4. Konfiguracja położenia bazy cytowań i bazy OSJ
Plik konfiguracyjny zawiera wskazanie położenia bazy cytowań. Przy standardowej instalacji są to wskazania:

```
<jndi:binding name="omega-psir/citationsPath">
    <jndi:value trim="true">/data/repo/ph/pw_db</jndi:value>
</jndi:binding>
<jndi:binding name="omega-psir/osjPath">
    <jndi:value trim="true">/data/repo/ph/OSJ_Ontology_103.xls</jndi:value>
</jndi:binding>
```

5. Konfiguracja zagnieżdżenia repozytorium
Oprócz tego, plik konfiguracyjny może zawierać parametr, który wskazuje adres strony zagnieżdżającej aplikację Repozytorium (podobnie jak strona bazy Wiedzy PW zagnieżdża stronę Repozytorium PW):

```
<jndi:binding name="omega-psir/nestingSiteUrl">
    <jndi:value trim="true">http://repo.bg.pw.edu.pl/index.php/pl/r</jndi:value>
</jndi:binding>
```

Wykorzystanie tego mechanizmu wymaga przygotowania odpowiedniej strony zagnieżdżającej, co wykracza poza zakres niniejszej instrukcji.

W przypadku bezpośredniego korzystania z aplikacji repozytorium parametr ten powinien być pusty:

```
<jndi:binding name="omega-psir/nestingSiteUrl">
    <jndi:value trim="true"></jndi:value>
</jndi:binding>
```

6. Konfiguracja domeny document.domain
   W przypadku zagnieżdżania odwołań do Repozytorium na innych stronach, plik konfiguracyjny powinien zawierać parametr domain wskazujący domenę, w ramach której wszystkie strony korzystają z Repozytorium. Oznacza to jednocześnie, że prawidłowo będą działały tylko te strony zagnieżdżające, które funkcjonują w ramach wskazanej domeny. Specyfikację dla Politechniki Warszawskiej pokazano poniżej:

```
<jndi:binding name="omega-psir/domain">
    <jndi:value trim="true">pw.edu.pl</jndi:value>
</jndi:binding>
```

7. Konfiguracja korzystania z portali społecznościowych
   Parametr addthisKey określa kod używany przy korzystaniu z portali społecznościowych. Mechanizm ten jest udostępniany pod adresem: http://www.addthis.com.

```
<jndi:binding name="omega-psir/addthisKey">
    <jndi:value trim="true"></jndi:value>
</jndi:binding>
```

8. Konfiguracja wykorzystania Google Analytics
   Parametr analyticsKey określa kod używany przy korzystywaniu z Google Analytics poziomu opisu publikacji z portalami społecznościowymi.

```
<jndi:binding name="omega-psir/analyticsKey">
    <jndi:value trim="true"></jndi:value>
</jndi:binding>
```

9. Konfiguracja dostępu do wykorzystywanego serwera pocztowego służącego do wymiany korespondencji poczty elektronicznej pomiędzy systemem Repozytorium a jego użytkownikami.
   Poniższe parametry konfiguracyjne wskazują serwer pocztowy oraz sposób dostępu zapewniający skuteczne wysyłanie wiadomości przez system Repozytorium w instalacji Politechniki Warszawskiej:

```
<jndi:binding name="omega-psir/mailServer">
    <jndi:value trim="true">elektron.elka.pw.edu.pl</jndi:value>
</jndi:binding>

<jndi:binding name="omega-psir/mailServerPort">
    <jndi:value trim="true">25</jndi:value>
</jndi:binding>

<jndi:binding name="omega-psir/mailServerUser">
    <jndi:value trim="true">repo</jndi:value>
</jndi:binding>

<jndi:binding name="omega-psir/mailServerPassword">
    <jndi:value trim="true"></jndi:value>
</jndi:binding>

<jndi:binding name="omega-psir/mailSenderAddress">
    <jndi:value trim="true">no-reply@repo.pw.edu.pl</jndi:value>
</jndi:binding>
```

10. Aktualnie wykorzystywane parametry konfiguracyjne można zweryfikować poprzez użycie funkcji administracyjnej dostępnej w menu administratora: systemConfiguration albo poprzez wpisanie odpowiedniego adresu: http://repo.pw.edu.pl/admin/systemConfiguration.seam?lang=pl

    Przykładowy raport dotyczący aktualnych parametrów konfiguracyjnych:

    Konfiguracja Wdrozenia

    instanceCode : PW
    idPrefix : WUT
    cas/casServerUrlPrefix : https://saturn.elka.pw.edu.pl/cas
    cas/casServerLoginUrl : https://saturn.elka.pw.edu.pl/cas/login
    cas/serverName : http://repo.pw.edu.pl
    citationsPath : /data/repo/ph/pw_db
    osjPath : /data/repo/ph/OSJ_Ontology_103.xls xls
    nestingSiteUrl :
    domain : pw.edu.pl
    addthisKey :
    analyticsKey :
    mailServer : elektron.elka.pw.edu.pl
    mailServerPort : 25
    mailServerUser : repo
    mailServerPassword :
    mailSenderAddress : no-reply@repo.pw.edu.pl
    repoPass :

9. Uruchomienie aplikacji systemu Repozytorium

1. Należy uruchomić serwer JBoss, np. uruchamiając go lokalnie za pomocą polecenia:

    /var/lib/jboss-4.2.2.GA/bin/run.sh –b localhost

Korzystając z lokalnego klienta webowego można sprawdzić, czy serwer JBoss udostępnia aplikację Repozytorium, np.

links http://localhost:8080/search.seam

W przypadku, gdy instalacja dotyczy startowej bazy danych, przy pierwszym wejściu do aplikacji, wykonywana jest czasochłonna operacja budowania indeksów i może zaistnieć potrzeba oczekiwania na odpowiedź serwera, ale stan ten można obserwować w logu serwera JBoss.

2. Zabezpieczenie serwera JBoss i wystawienie aplikacji poza localhost z wykorzystaniem serwera Apache
   - Zalecane jest wyłączenie administracyjnych modułów JBoss, np. web_console, jmx-console, itp. (dodatkowo można zablokować dostęp do tych modułów na poziomie konfiguracji serwera Apache - RewriteRule)

   - Zastosowanie modułu mod_jk w celu wykorzystania serwera Apache jako proxy do serwera JBoss.
     Należy zapewnić możliwość korzystania z modułu mod_jk i następnie odpowiednio go skonfigurować. Przykładowe specyfikacje dotyczące takiej konfiguracji:
     - mod_jk.conf
       ```
       JkWorkersFile     /etc/apache2/workers.properties
       JkLogFile         /var/log/apache2/mod_jk.log
       JkLogLevel        info
       JkMount /* worker1
       ```

     - mod_jk.load
       ```
       LoadModule jk_module /usr/lib/apache2/modules/mod_jk.so
       ```

     - workers.properties
       ```
       workers.properties
       worker.list=worker1
       worker.worker1.type=ajp13
       worker.worker1.host=localhost
       worker.worker1.port=8009
       [uri:/*]
       ```

   - Zablokowanie dostępu do serwera JBoss poza protokołem AJP (używanym przez mod_jk). Specyfikacje związane z konfiguracją tych parametrów serwera JBoss znajdują się w pliku /var/lib/jboss-4.2.2.GA/server/default/deploy/jboss-web.deployer/server.xml
     Należy wyłączyć (wykomentować) wszystkie specyfikacje connector poza AJP 1.3 Connector.

- Dodatkowo w pliku server.xml należy zmodyfikować specyfikacje AJP 1.3 Connector, aby zapewnić prawidłową transmisję polskich znaków i kompresji danych.
  Parametry te określone są w definicji AJP 1.3 Connector:

```
<Connector port="8009" address="${jboss.bind.address}"
protocol="AJP/1.3" emptySessionPath="true" enableLookups="false"
redirectPort="8443" useBodyEncodingForURI="true" URIEncoding="UTF-
8" compression="off" compressionMinSize="2048"
compressableMimeType="text/html,text/xml,text/plain,text/css,text/
javascript"/>
```

  Zasadniczo, można byłoby zastosować tutaj kompresję, ale bardziej wskazane jest wyłączenie kompresji w konektorze, przy jednoczesnym zapewnieniu kompresji na poziomie serwera www. Odpowiednia konfiguracja serwera apache powinna wyglądać następująco:
  - włączenie modułu kompresji:
  ```
  LoadModule deflate_module modules/mod_deflate.so
  ```
  - moduł kompresji wymaga zainstalowania modułu mod_filter (względnie mod_ext_filter):
  ```
  LoadModule filter_module modules/mod_filter.so
  ```

  - specyfikacja kompresji:
  ```
  AddOutputFilterByType DEFLATE text/html text/plain text/xml
  AddOutputFilterByType DEFLATE text/css
  AddOutputFilterByType DEFLATE appliation/x-javascript
  application/javascript application/ecmascript
  AddOutputFilterByType DEFLATE application/rss+xml
  ```

3. Dodatkowa konfiguracja serwera JBoss

   W celu ograniczenia objętości logów generowanych przez serwer JBoss należy dobrać wartości odpowiednich parametrów konfiguracyjnych w taki sposób, aby zawartość generowanych logów odpowiadała rzeczywistym potrzebom. Aktualnie wykorzystywane są poniższe ustawienia w pliku /var/lib/jboss-4.2.2.GA/server/default/conf/jboss-log4j.xml:

```
<category name="pl.edu.pw.ii">
    <priority value="INFO"/>
</category>
<root>
    <priority value="ERROR" />
    <appender-ref ref="CONSOLE"/>
    <appender-ref ref="FILE"/>
</root>
```

4. Skopiowanie właściwych bibliotek JSF używanych przez serwer JBoss.

   W celu zapewnienia prawidłowego działania aplikacji serwer JBoss musi korzystać z określonych wersji bibliotek JSF. Biblioteki te znajdują się w katalogu /var/lib/jboss-4.2.2.GA/server/default/deploy/jboss-web.deployer/jsf-libs:

   jsf-imp.jar

   jsf-api.jar

   Aplikacja repozytorium wymaga stosowania tych bibliotek w wersji 1.2_14-b01-FCS.

   Jeśli zainstalowana wersja serwera JBoss korzysta z innych wersji tych bibliotek, należy podmienić zainstalowane wersje dostarczonymi plikami umieszczonymi w katalogu aplikacji /data/repo/jsf-libs

10. Przygotowanie baz systemowych i pomocniczych

W przypadku, gdy instalacja dotyczy pustej bazy danych, należałoby w niej utworzyć elementy umożliwiające prawidłowe korzystanie z modułu automatycznego nadawania punktów publikacjom podlegającym ocenie ministerialnej (scoring.drl) oraz algorytmy oceny ekspertów (expertscoretotal.drl). Wstępnie można załączyć do systemu algorytmy punktacji z katalogu repo-init aplikacji. Do tego celu służą funkcje administracyjne Algorytmy punktacji oraz Algorytmy oceny eksperta. Trzeba pamiętać, że załadowane do Repozytorium algorytmy nie powinny być oznakowane jako służące do symulacji.

Ponadto, należy zainicjować inne elementy wspierające korzystanie z aplikacji: bazę czasopism, bazę konferencji (wykaz tytułów i wykaz wydarzeń), bazę instytucji, języki, typy projektów itp. Pakiet instalacyjny zawiera katalog z plikami w formacie XML, które mogą być wykorzystane do zaimportowania wybranych danych. Importowanie plików XML dostępne jest administratorowi systemu repozytorium w ramach korzystania z funkcji pomocniczej Import. Przygotowane dane występują jako rekordy PW i posiadają oryginalne identyfikatory, ale nie powinno to przeszkadzać w ich użytkowaniu, wręcz przeciwnie takie podejście będzie sprzyjać wygodnej aktualizacji tych danych.

Aktualnie wykorzystywane zasoby walidacyjne i pomocnicze oraz liczbę zdefiniowanych użytkowników w Repozytorium można zweryfikować poprzez użycie funkcji administracyjnej dostępnej w menu administratora: systemConfiguration albo poprzez wpisanie odpowiedniego adresu:
http://repo.pw.edu.pl/admin/systemConfiguration.seam?lang=pl

Przykładowy raport dotyczący aktualnego stanu wymaganych danych systemowych i pomocniczych:

Konfiguracja Repozytorium

Algorytm punktacji dorobku (bez flagi tylko do symulacji):10
Afiliacja nadrzedna o polskim akronimie identycznym z parametrem wdrozenia instanceCode:1
Lista czasopism:18714
Aktywny algorytm oceny eksperta:1
Uzytkownicy:197

Oprócz przygotowania systemowych danych walidacyjnych i pomocniczych, przed rozpoczęciem wprowadzania danych publikacji, projektów itp., należy utworzyć w Repozytorium strukturę jednostek uczelni, a także załadować dane pracowników przechowywane jako opis autorów.

11. Uwagi końcowe

1. Na zakończenie procesu instalowania należy zwrócić specjalną uwagę na to, czy w katalogach aplikacji Repozytorium oraz katalogu serwera JBoss właścicielem wszystkich plików jest dedykowany użytkownik repo.

2. W celu ograniczenia obciążenia systemu przez roboty internetowe wskazane jest przygotowanie odpowiedniego pliku robots.txt, który powinien być umieszczony w katalogu DocumentRoot serwera Apache. Aby plik ten był respektowany w konfiguracji serwera Apache związanej z przekazywaniem odwołań do modułu mod_jk trzeba wyłączyć przekazywanie odwołania do pliku robots.txt, np.

```
# allow robots.txt to be served by apache (/var/www/html) ( exclude from
transfering to mod_jk )
SetEnvIf Request_URI "/robots.txt" no-jk
```

Przykładowy plik robots.txt:

```
User-agent: *                   # all robots
Disallow: /stats/               # disallow on this directory
Disallow: /SearchExpert.seam*   # disallow these requests
Disallow: /SearchGlobal.seam*   # disallow these requests
Crawl-delay: 10                 # wait 10 sec with next request
```

12. Wnioski i dalsze kroki

Po zainstalowaniu systemu, zalecane jest utworzenie dokumentacji po instalacyjnej, w której należy opisać wykonaną instalację, w tym między innymi:

- położenie poszczególnych elementów oprogramowania i baz danych
- zastosowane parametry konfiguracyjne