

# Rozdział 8

## Baza danych uczących

Adam CHOLEWA

### 8.1. Wstęp

Procesy pozyskiwania wiedzy dla systemów doradczych oraz procesy weryfikacji tych systemów wymagają danych uczących, które mogą być stosowane jako dane trenujące lub testujące. W chwili obecnej źródła danych stosują różne formaty danych. Utrudnia to stosowanie tych danych i ogranicza możliwości automatyzacji procesów pozyskiwania zawartej w nich wiedzy.

W niniejszym rozdziale opisano Bazę Danych Uczących (BDU) pozwalającą na przechowywanie w jednolity sposób, danych pochodzących z różnych źródeł, w których pierwotnie były one zapisane w różnych formatach. Baza ta jest integralną częścią systemu DiaDyn.

Do pełnego przedstawienia znaczenia danych, niezbędnego dla właściwego ich zastosowania, potrzebne są:

- informacje o obiekcie, którego dotyczą dane,
- informacje o rozpatrywanych cechach sygnałów,
- wartości cech sygnałów.

Zdecydowano, że wszystkie dane będą przekazywane do BDU w postaci odpowiednio przygotowanych plików XML. W celu przekazania kompletu informacji wprowadzono cztery rodzaje plików XML. W plikach tych będą przekazane następujące dane:

- informacje o obiekcie,
- informacje o rozpatrywanych cechach sygnałów,
- informacje o konfiguracji wektora danych,
- wektor danych.

### 8.2. Opis zastosowanego rozwiązania

W celu umożliwienia przekazywania danych z różnych źródeł do BDU DiaDyn oraz w celu umożliwienia pobierania zasobów tej BDU opracowano zestawy bibliotek i procedur, odpowiednio o nazwach „Zapis do BDU” oraz „Odczyt z BDU”.

#### 8.2.1. Formaty plików

Dane są przekazywane do BDU w postaci odpowiednio przygotowanych plików. Założono, że przekazywane pliki będą zapisane w formacie XML. Dzięki takiemu rozwiązaniu posiadają one jednolitą

składnię i w łatwy sposób można sprawdzać formalną poprawność ich treści. W procesie przesyłania plików z danymi do BDU zastosowany jest dedykowany walidator napisany w języku PHP, który pozwala na wykrywanie niezgodności formatu plików z ich specyfikacją.

Założono, że dane uczące pochodzące z dowolnego źródła będą wprowadzane do BDU w postaci następujących czterech plików:

- plik określający obiekt, z którym związane są dane,
- plik określający cechy sygnałów, których wartości przekazywane są do BDU,
- plik określający postać wektorów wartości cech sygnałów (w pliku tym określa się uporządkowanie cech w wektorze cech),
- plik z wektorami cech.

Wymagane jest przekazywanie plików do BDU w podanej kolejności.

Przekazywane pliki posiadają unikalne identyfikatory umożliwiające zapisywanie relacji występujących między tymi plikami. Założono, że zarządzanie identyfikatorami zapewniające ich unikalność będzie realizowane w ten sposób, iż każdemu zespołowi autorów zasobów BDU przydzielony będzie odpowiedni przedział wartości tych identyfikatorów.

W celu zapisania treści wymaganych plików niezbędna jest znajomość ich składni. Plik definiujący obiekt bądź grupę obiektów musi mieć następującą postać jak na Wydruk 8.1.

Wydruk 8.1: Postać pliku definiującego obiekt badań

```

1 <?xml version="1.0" encoding="ISO-8859-2" ?>
2 <!DOCTYPE objects SYSTEM "_objects.dtd">
3 <objects>
4   <version>
5     <ver_objects_id></ver_objects_id>
6     <ver_date></ver_date>
7     <ver_time></ver_time>
8     <ver_organisation></ver_organisation>
9     <ver_createdby></ver_createdby>
10    <ver_info_short></ver_info_short>
11    <ver_info_long></ver_info_long>
12    <ver_comment></ver_comment>
13  </version>
14  <object>
15    <object_id></object_id>
16    <name></name>
17    <location></location>
18    <info_short></info_short>
19    <info_long></info_long>
20    <comment></comment>
21  </object>
22  ...
23 </objects>

```

W każdym z rozpatrywanych plików XML występuje nagłówek. Nagłówki mają bardzo zbliżoną postać. W nagłówku wskazywany jest przede wszystkim sposób kodowania znaków w dokumencie. Przyjęto, że jest to ISO-8859-2 (Wydruk 8.1). Określono również nazwę pliku DTD (Document Type Definition) zawierającego definicję poprawnej składni pliku XML. W każdym pliku zapisana jest informacja o jego wersji obejmująca:

- identyfikator pliku z definicją/definicjami obiektów (ver\_objects\_id) np. abc0003,
- datę zdefiniowania pliku (ver\_date) np. 2007-10-24,
- czas (ver\_time) np. 20:34,
- instytucję lub jednostkę, która zapisała plik (ver\_organisation) np. Politechnika Śląska,
- informację o osobie, która zapisała plik lub była odpowiedzialna za jego zapisanie (ver\_createdby) np. Jan Kowalski,

- informacje dodatkowe w postaci skróconej (ver\_info\_short),
- informacje dodatkowe w postaci pełnej (ver\_info\_long),
- komentarz (ver\_comment).

W pliku z informacjami o obiekcie należy poza nagłówkiem wpisać także dane dotyczące obiektu lub grupy obiektów, obejmujące:

- identyfikator obiektu (object\_id) np. 004
- nazwę obiektu (name) np. turbospół 13K215
- lokalizację obiektu (location) np. Kozienice
- skróconą informację o obiekcie (info\_short) np. turbospół z turbiną 13K215
- pełną informację o obiekcie (info\_long)
- komentarz (comment)

Wydruk 8.2: Postać pliku definiującego cechy sygnałów

```

1 <?xml version="1.0" encoding="ISO-8859-2" ?>
2 <!DOCTYPE features SYSTEM "_features.dtd">
3 <features>
4   <version>
5     <ver_features_id></ver_features_id>
6     <ver_date></ver_date>
7     <ver_time></ver_time>
8     <ver_organisation></ver_organisation>
9     <ver_createdby></ver_createdby>
10    <ver_info_short></ver_info_short>
11    <ver_info_long></ver_info_long>
12    <ver_comment></ver_comment>
13  </version>
14  <feature>
15    <feature_id></feature_id>
16    <name></name>
17    <unit></unit>
18    <dim></dim>
19    <info_short></info_short>
20    <info_long></info_long>
21    <comment></comment>
22  </feature>
23  ...
24 </features>

```

Kolejnym plikiem jaki należy przekazać do BDU jest plik zawierający definicje uwzględnianych cech sygnałów (Wydruk 8.2). W pliku tym nagłówek oraz fragment związany z określeniem wersji pliku mają taką samą postać jak w przypadku pliku pokazanego na Wydruk 8.1. W dalszej części pliku należy zapisać informacje o stosowanych cechach sygnałów:

- identyfikator cechy sygnału (feature\_id) np. 007,
- nazwa cechy (name) np. drgl7,
- użyta skala (unit) np. 0.1,
- jednostka podstawowa (dim) np. mm/s,
- skróconą informację o cechach (info\_short) np. Drgania lozyska, 7
- pełną informację o cechach (info\_long) np. Maksymalna amplituda predkosci drgan obudowy lozyska 7 w kierunku pionowym,
- komentarz (comment).

Wydruk 8.3: Plik definiujący postaci wektorów cech sygnałów

```

1 <?xml version="1.0" encoding="ISO-8859-2" ?>
2 <!DOCTYPE configs SYSTEM "_config.dtd">
3 <configs>
4   <version>
5     <ver_configs_id></ver_configs_id>
6     <ver_date></ver_date>
7     <ver_time></ver_time>
8     <ver_organisation></ver_organisation>
9     <ver_createdby></ver_createdby>
10    <ver_info_short></ver_info_short>
11    <ver_info_long></ver_info_long>
12    <ver_comment></ver_comment>
13  </version>
14  <config>
15    <config_id></config_id>
16    <features_id></features_id>
17    <feature_id_vector></feature_id_vector>
18    <comment></comment>
19  </config>
20  ...
21 </configs>

```

Następnym z przekazywanych plików jest plik definiujący postaci wektorów cech. Podobnie jak w przypadku Wydruku 8.1 oraz Wydruku 8.2 plik pokazany na Wydruku 8.3 posiada taką samą postać nagłówka. W dalszej jego części zdefiniowano postaci wektorów cech przekazywanych w pliku z danymi. Na tę część pliku składają się :

- identyfikator konfiguracji (`config_id`) np. 002,
- identyfikator zestawu opisującego cechy - jest to wartość wpisywana w nagłówku pliku z definicją cech (`features_id`) np. abc009,
- wektor składający się z identyfikatorów cech zapisanych w pliku, na który wskazuje identyfikator (`features_id`) np. 003 002 008 009 007 010 012.

Dopuszcza się możliwość definiowania wektorów jednoelementowych. Należy zauważyć, że w pliku z definicjami wektorów można użyć definicji zestawów cech zapisanych przez inny zespół wprowadzający dane (numer zespołu wprowadzającego dane jest rozpoznawany na podstawie identyfikatorów `objects_id`, `features_id` oraz `configs_id`).

Wydruk 8.4: Plik z danymi

```

1 <?xml version="1.0" encoding="ISO-8859-2" ?>
2 <!DOCTYPE vectors SYSTEM "_vectors.dtd">
3 <vectors>
4   <version>
5     <ver_vectors_id></ver_vectors_id>
6     <ver_object_id></ver_object_id>
7     <ver_config_id></ver_config_id>
8     <ver_date></ver_date>
9     <ver_time></ver_time>
10    <ver_organisation></ver_organisation>
11    <ver_createdby></ver_createdby>
12    <ver_info_short></ver_info_short>
13    <ver_info_long></ver_info_long>
14    <ver_comment></ver_comment>
15  </version>
16  <vector>
17    <vector_id></vector_id>
18    <date></date>
19    <time></time>
20    <values></values>

```

```

21         <comment></comment>
22     </vector>
23     ...
24 </vectors>

```

Ostatnim z przekazywanych plików jest plik z danymi pobranymi z obiektu. Plik ten jako jedyny ma inną postać nagłówka. W nagłówku oprócz identyfikatora pliku z danymi (`ver_vectors_id`) określono:

- jakiego obiektu dotyczą dane (`ver_objects_id`)
- jaki plik konfiguracyjny definiuje postaci wektorów (`ver_configs_id`)

W ostatniej części pliku należy podać informacje na temat:

- identyfikatora wektora (`vector_id`),
- daty której dotyczą dane (`date`),
- czasu (`time`),
- wektora wartości cech (`values`),
- komentarza (`comment`).

We wszystkich definiowanych plikach pola `comments` oraz pola `info_long` nie są polami wymaganymi.

### 8.2.2. Schemat bazy danych uczących

Podczas przekazywania plików do BDU część danych zamieszczonych w plikach jest zapisywana w relacyjnej bazie danych założonej w serwerze baz danych MySQL, a pliki z danymi kopiowane są do BDU. Relacyjna baza danych składa się z następujących 12 tablic:

- `config` (Wydruk 8.5) - tablica ta przechowuje informacje o dostępnych konfiguracjach zestawów danych,
- `config_sets` (Wydruk 8.6) - tablica ta przechowuje informacje na temat tego w jaki sposób pogrupowane zostały zestawy konfiguracyjne,
- `configs` (Wydruk 8.7) - tablica ta przechowuje informacje na temat tego w jakim pliku zapisano konkretny zestaw konfiguracyjny,
- `feature` (Wydruk 8.8) - tablica ta przechowuje informacje o dostępnych cechach,
- `feature_sets` (Wydruk 8.9) - tablica ta przechowuje informacje o możliwych zestawach cech,
- `features` (Wydruk 8.10) - tablica ta przechowuje informacje na temat tego w jakim pliku zapisano konkretny zestaw cech,
- `object` (Wydruk 8.11) - tablica ta przechowuje informacje o dostępnych obiektach,
- `object_sets` (Wydruk 8.12) - tablica ta przechowuje informacje o możliwych grupach obiektów,
- `objects` (Wydruk 8.13) - tablica ta przechowuje informacje na temat tego w jakim pliku zapisano konkretną grupę obiektów,
- `vector` (Wydruk 8.14),
- `vector_sets` (Wydruk 8.15),
- `vectors` (Wydruk 8.16).

Relacyjną bazę danych przygotowano dla serwera baz danych MySQL.

Wydruk 8.5: Definicja tablicy `config`

```

1 CREATE TABLE 'config' (
2   config_id varchar(10) NOT NULL default '',
3   features_id varchar(10) NOT NULL default '',
4   feature_id_vector varchar(45) NOT NULL,
5   comment varchar(255) NOT NULL default '',
6   PRIMARY KEY (config_id)
7 )

```

Wydruk 8.6: Definicja tablicy config\_sets

```

1 CREATE TABLE config_sets (
2 id int(10) unsigned NOT NULL auto_increment,
3 configs_id varchar(10) NOT NULL default '',
4 config_id varchar(10) NOT NULL default '',
5 PRIMARY KEY (id)
6 )

```

Wydruk 8.7: Definicja tablicy configs

```

1 CREATE TABLE configs (
2 configs_id varchar(10) default NULL,
3 date varchar(45) default '-',
4 time varchar(45) default '-',
5 organisation varchar(45) default '-',
6 createdby varchar(45) default NULL,
7 info_short varchar(255) default '-',
8 info_long varchar(255) default '-',
9 comment varchar(255) default NULL,
10 fileno int(10) unsigned NOT NULL default '0',
11 filename varchar(255) NOT NULL default '',
12 a_id int(10) unsigned NOT NULL auto_increment,
13 PRIMARY KEY (a_id)
14 )

```

Wydruk 8.8: Definicja tablicy feature

```

1 CREATE TABLE feature (
2 feature_id varchar(10) NOT NULL default '',
3 name varchar(45) NOT NULL default '',
4 unit varchar(45) NOT NULL default '',
5 dim varchar(45) NOT NULL default '',
6 info_short varchar(255) NOT NULL default '',
7 info_long varchar(255) NOT NULL default '',
8 comment varchar(255) NOT NULL default '',
9 PRIMARY KEY (feature_id)
10 )

```

Wydruk 8.9: Definicja tablicy feature\_sets

```

1 CREATE TABLE feature_sets (
2 id int(10) unsigned NOT NULL auto_increment,
3 features_id varchar(10) NOT NULL default '',
4 feature_id varchar(10) NOT NULL default '',
5 PRIMARY KEY (id)
6 )

```

Wydruk 8.10: Definicja tablicy features

```

1 CREATE TABLE features (
2 features_id varchar(10) NOT NULL default '',
3 date varchar(45) default '-',
4 time varchar(45) default '-',
5 organisation varchar(45) default '-',
6 createdby varchar(45) default NULL,
7 info_short varchar(255) default '-',
8 info_long varchar(255) default '-',
9 comment varchar(255) default NULL,
10 fileno int(10) unsigned NOT NULL default '0',
11 filename varchar(255) NOT NULL default '',
12 id int(10) unsigned NOT NULL auto_increment,

```

```

13 PRIMARY KEY (id)
14 )

```

Wydruk 8.11: Definicja tablicy object

```

1 CREATE TABLE object (
2 object_id varchar(10) NOT NULL default '',
3 name varchar(255) default NULL,
4 info_short varchar(255) default NULL,
5 info_long varchar(255) default NULL,
6 location varchar(45) NOT NULL,
7 comment varchar(255) NOT NULL default '',
8 PRIMARY KEY (object_id)
9 )

```

Wydruk 8.12: Definicja tablicy object\_sets

```

1 CREATE TABLE object_sets (
2 id int(10) unsigned NOT NULL auto_increment,
3 objects_id varchar(10) NOT NULL default '',
4 object_id varchar(10) NOT NULL default '',
5 PRIMARY KEY (id)
6 )

```

Wydruk 8.13: Definicja tablicy objects

```

1 CREATE TABLE objects (
2 objects_id varchar(10) NOT NULL default '',
3 date varchar(45) default NULL,
4 time varchar(45) default NULL,
5 organisation varchar(45) default NULL,
6 createdby varchar(45) default NULL,
7 info_short varchar(255) default NULL,
8 info_long varchar(255) default NULL,
9 comment varchar(255) default NULL,
10 fileno int(10) unsigned NOT NULL default '0',
11 filename varchar(255) NOT NULL,
12 id int(10) unsigned NOT NULL auto_increment,
13 PRIMARY KEY (id)
14 )

```

Wydruk 8.14: Definicja tablicy vector

```

1 CREATE TABLE vector (
2 vector_id varchar(10) NOT NULL default '',
3 date1 varchar(45) NOT NULL default '',
4 time1 varchar(45) NOT NULL default '',
5 values1 varchar(255) NOT NULL default '',
6 comment1 varchar(255) NOT NULL default '',
7 PRIMARY KEY (vector_id)
8 )

```

Wydruk 8.15: Definicja tablicy vector\_sets

```

1 CREATE TABLE vector_sets (
2 id int(10) unsigned NOT NULL auto_increment,
3 vectors_id varchar(10) NOT NULL default '',
4 vector_id varchar(10) NOT NULL default '',
5 PRIMARY KEY (id)
6 )

```

Wydruk 8.16: Definicja tablicy vectors

```

1 CREATE TABLE vectors (
2 vectors_id varchar(10) NOT NULL default '',
3 date varchar(45) default '-',
4 time varchar(45) default '-',
5 organisation varchar(45) default '-',
6 createdby varchar(45) default NULL,
7 info_short varchar(255) default '-',
8 info_long varchar(255) default '-',
9 comment varchar(255) default NULL,
10 fileno int(10) unsigned NOT NULL default '0',
11 filename varchar(255) NOT NULL default '',
12 object_id varchar(10) NOT NULL default '',
13 config_id varchar(10) NOT NULL default '',
14 id int(10) unsigned NOT NULL auto_increment,
15 PRIMARY KEY (id)
16 )

```

### 8.2.3. Scenariusz przekazywania danych do BDU

Przekazanie danych do BDU DiaDyn odbywa się wg następującego scenariusza:

- wskazywany jest rodzaj przekazywanych danych, gdzie możliwe jest wskazanie jednego z czterech wcześniej określonych typów przekazywanych danych (informacje o obiekcie, definicje cech sygnałów, konfiguracje, wektory wartości cech sygnałów),
- po wyborze typu danych i wyborze pliku XML zawierającego te dane jest on wczytywany do BDU,
- podczas wczytywania pliku następuje walidacja jego treści polegająca między innymi na sprawdzeniu czy wszystkie wymagane informacje zostały w nim zawarte,
- dla pliku o poprawnej treści sprawdzana jest unikalność występujących w nim identyfikatorów, które nie mogą występować wcześniej w BDU,
- jeśli występuje powtórzenie identyfikatorów w treści pliku i treści danych wpisanych już do bazy to użytkownik zostaje o tym poinformowany odpowiednim komunikatem
- jeżeli nie występują takie powtórzenia to dane są wprowadzane do BDU,
- po zakończeniu procesu wprowadzania danych użytkownik otrzymuje odpowiedni komunikat.

### 8.2.4. Scenariusz odczytu danych z BDU

Odczyt danych z BDU odbywa się według następującego scenariusza:

- wskazywany jest rodzaj odczytywanych danych (podobnie jak w przypadku zapisu danych do BDU można przeszukiwać informacje o obiektach, o cechach sygnałów o konfiguracjach lub wektorach wartości cech),
- wskazywany jest rodzaj szukanych informacji (możliwe jest przeszukiwanie np. na podstawie informacji o autorze pliku z danymi, na podstawie daty itp.),
- sformułowane jest zapytanie z użyciem składni języka SQL w celu wyszukiwania danych,
- wyświetlana jest informacja o zasobach spełniających podane warunki.

Dane z BDU pobierane są w postaci pliku XML.



### 8.2.5. Przekazywanie danych z BDU do środowiska MATLAB

Pliki XML pobrane z BDU DiaDyn mogą być importowane w środowisku MATLAB. Do importu należy użyć funkcji `xml_read()`. Funkcja ta wywołuje funkcję `xmlread` Matlaba, a następnie konwertuje dane wyjściowe tej funkcji na drzewo struktur Matlaba. Funkcja `xml_read` jest dostępna dla Matlaba począwszy od wersji R2006b. Można ją pobrać ze strony: <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=12907&objectType=file>.

## 8.3. Przykład zastosowania

### 8.3.1. Przygotowanie pliku z danymi wejściowymi do BDU

Pliki zawierające dane przeznaczone do zapisania w BDU należy przygotować zgodnie z opisem przedstawionym w rozdziale 8.2.1.

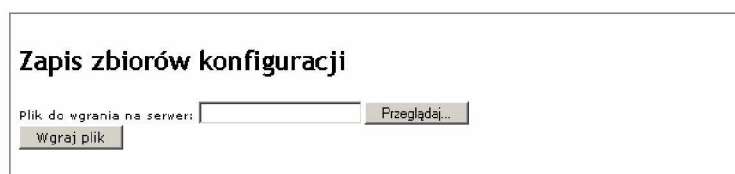
### 8.3.2. Przykład przekazania pliku do BDU

Pliki z danymi są przesyłane do BDU DiaDyn za pośrednictwem przeglądarki WWW, według zestawu procedur i funkcji opracowanych w języku PHP. Podczas przesyłania danych zawartość każdego pliku jest walidowana. Jednocześnie uzupełniane są tablice zawierające opisy plików w bazie danych. W celu przesłania odpowiednio przygotowanego pliku lub zestawu plików do BDU DiaDyn należy w lewym panelu okna systemu DiaDyn odszukać pozycję „Zapis do BDU”. Pojawi się okno jak na rys. 8.1.



Rys. 8.1: Postać okna głównego po wyborze opcji „Zapis do BDU”

Możliwe jest przesłanie plików zawierających definicje zbioru obiektów, zbioru sygnałów, zbioru konfiguracji oraz zbioru danych. Po wybraniu odpowiedniej opcji pojawi się okno (Rys. 8.2) umożliwiające wskazanie na dysku lokalnym użytkownika odpowiedniego pliku xml zawierającego przesyłany zbiór.



Rys. 8.2: Postać okna umożliwiającego wybór i przesłanie pliku do BDU

Jeśli plik posiada prawidłową strukturę oraz brak w nim błędów i w BDU nie występuje jeszcze plik o takim samym identyfikatorze, to uzupełniana jest zawartość BDU i pojawia się okno z podsumowaniem operacji przekazywania pliku (Rys. 8.3).



Rys. 8.3: Postać okna podsumowującego przesłanie pliku do BDU DiaDyn

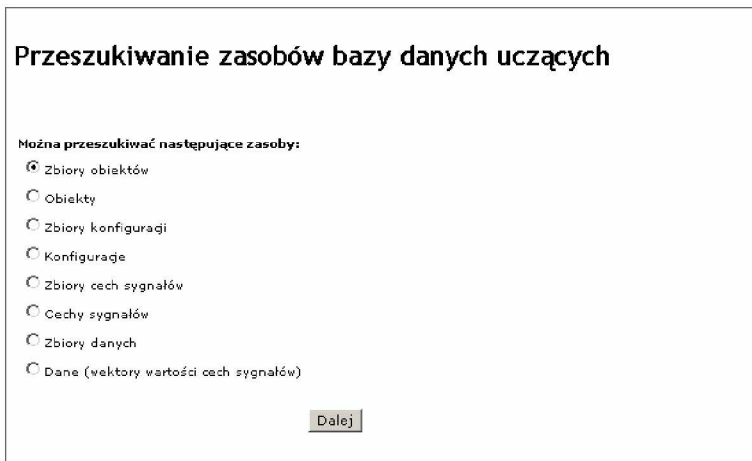
Jeżeli w BDU DiaDyn znajduje się już wcześniej zapisany plik z identyfikatorem określonym w nagłówku przesyłanego pliku (rozdział 8.2.1) to zostaje wyświetlony komunikat jak na Rys. 8.4.



Rys. 8.4: Postać okna informującego o wystąpieniu błędu

### 8.3.3. Przykład wyszukania i pobrania pliku z BDU

Zawartość BDU DiaDyn może być przeszukiwana i odczytywana za pośrednictwem przeglądarki WWW według zestawu procedur i funkcji opracowanych w języku PHP. W celu przeszukania BDU oraz odczytu pliku zawierającego poszukiwane dane należy w lewym panelu okna systemu DiaDyn odszukać pozycję „Odczyt z BDU”. Pojawi się okno jak na Rys. 8.5.

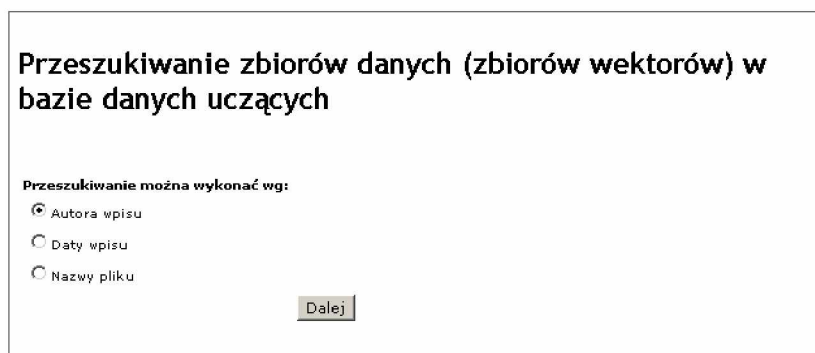


Rys. 8.5: Postać okna głównego po wyborze opcji „Odczyt z BDU”

Po wyborze rodzaju przeszukiwanych zasobów następuje przejście do okna umożliwiającego określenie warunku wyszukiwania. Zasoby zawierające definicje zbiorów obiektów, zbiorów konfiguracji, zbiorów cech sygnałów lub zbiorów danych mogą być przeszukiwane na podstawie autora wpisu, daty wpisu lub nazwy pliku (Rys. 8.6).

Zasoby zawierające definicje obiektów mogą być przeszukiwane na podstawie nazwy obiektu, lokalizacji obiektu lub znanych informacji o obiekcie (Rys. 8.7).

Zasoby zawierające definicje cech sygnałów mogą być przeszukiwane na podstawie nazwy cechy sygnału, skali, jednostki lub komentarza (Rys. 8.8).



**Przeszukiwanie zbiorów danych (zbiorów wektorów) w bazie danych uczących**

Przeszukiwanie można wykonać wg:

Autora wpisu

Daty wpisu

Nazwy pliku

Dalej

Rys. 8.6: Postać okna pozwalającego na wybór wyszukiwanej informacji



**Przeszukiwanie obiektów w bazie danych uczących**

Przeszukiwanie można wykonać wg:

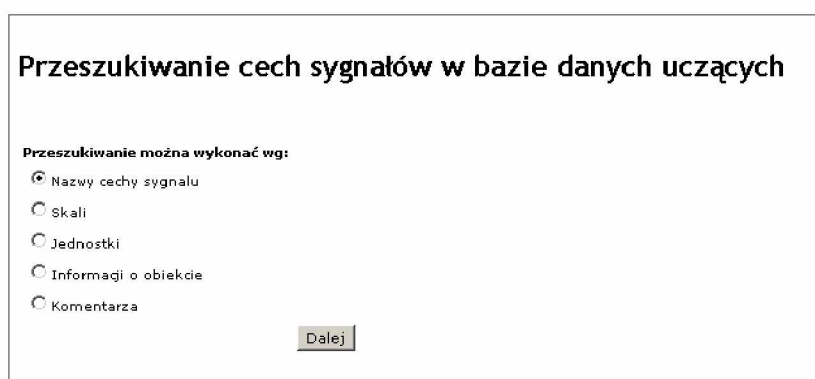
Nazwy obiektu

Informacji o obiekcie

Lokalizacji obiektu

Dalej

Rys. 8.7: Postać okna z wyborem wyszukiwanej informacji o obiekcie



**Przeszukiwanie cech sygnałów w bazie danych uczących**

Przeszukiwanie można wykonać wg:

Nazwy cechy sygnału

Skali

Jednostki

Informacji o obiekcie

Komentarza

Dalej

Rys. 8.8: Postać okna z wyborem wyszukiwanej informacji o cechach sygnałów

Kolejnym krokiem jest określenie uściślenie warunku wyszukiwania (Rys. 8.9), zapisywanego z zastosowaniem składni języka SQL. Przykładowe warunki wyszukiwania pola o określonej treści mogą być następujące:

- % - akceptuje dowolną treść pola,
- Bartosz% - akceptuje pola, których treść rozpoczyna się od słowa „Bartosz” ,
- %g% - akceptuje pola, w których treści występuje znak „g”

Po zakończeniu wyszukiwania według zadanych warunków wyświetlana jest lista zasobów spełniających te warunki (Rys. 8.10). Na tej liście, w ostatniej kolumnie, wskazywane są pliki zawierające poszukiwane dane. Kliknięcie na wybrane wskazanie pliku umożliwi zapisanie znalezionej zasobu na dysku lokalnym.

### Przeszukiwanie obiektów w bazie danych uczących

Warunek wyszukiwania dotyczący pola opisującego **nazwę obiektu**:

Podczas wpisywania warunku wyszukiwania należy posługiwać się składnią języka SQL.

Przykłady warunków wyszukiwania rozpatrywanego pola:

- % - akceptuje dowolną treść pola,
- Bartosz% - akceptuje pola, których treść rozpoczyna się od słowa 'Bartosz',
- %g% - akceptuje pola, w których treści występuje znak 'g'

Rys. 8.9: Postać okna z możliwością określenia warunku wyszukiwania

### Wynik przeszukiwania obiektów w bazie danych uczących

Wynik przeszukiwania uwzględni wyłącznie te elementy bazy danych, które odpowiadają podanemu wcześniej warunkowi wyszukiwania.

Nazwa obiektu	Opis	Streszczenie	Lokalizacja obiektu	Nazwa pliku do pobrania
Rotor Kit	Stanowisko Rotor Kit	-	Konarskiego 18a, 44-100 Gliwice	_objects_demo.xml

Rys. 8.10: Postać okna z wynikami wyszukiwania

## Bibliografia

- [8.1] AB MySQL. Mysql 5.1 reference manual. 2007.
- [8.2] Team The DragonFly Documentation. The dragonfly documentation project. *DragonFly Handbook*, 2006.